

Gestión inteligente de información y reconocimiento de patrones

Research in Computing Science

Series Editorial Board

Editors-in-Chief:

Grigori Sidorov (Mexico)
Gerhard Ritter (USA)
Jean Serra (France)
Ulises Cortés (Spain)

Associate Editors:

Jesús Angulo (France)
Jihad El-Sana (Israel)
Alexander Gelbukh (Mexico)
Ioannis Kakadiaris (USA)
Petros Maragos (Greece)
Julian Padget (UK)
Mateo Valero (Spain)

Editorial Coordination:

María Fernanda Ríos Zacarías

Research in Computing Science es una publicación trimestral, de circulación internacional, editada por el Centro de Investigación en Computación del IPN, para dar a conocer los avances de investigación científica y desarrollo tecnológico de la comunidad científica internacional. **Volumen 93**, mayo 2015. Tiraje: 500 ejemplares. *Certificado de Reserva de Derechos al Uso Exclusivo del Título* No. : 04-2005-121611550100-102, expedido por el Instituto Nacional de Derecho de Autor. *Certificado de Licitud de Título* No. 12897, *Certificado de licitud de Contenido* No. 10470, expedidos por la Comisión Calificadora de Publicaciones y Revistas Ilustradas. El contenido de los artículos es responsabilidad exclusiva de sus respectivos autores. Queda prohibida la reproducción total o parcial, por cualquier medio, sin el permiso expreso del editor, excepto para uso personal o de estudio haciendo cita explícita en la primera página de cada documento. Impreso en la Ciudad de México, en los Talleres Gráficos del IPN – Dirección de Publicaciones, Tres Guerras 27, Centro Histórico, México, D.F. Distribuida por el Centro de Investigación en Computación, Av. Juan de Dios Bátiz S/N, Esq. Av. Miguel Othón de Mendizábal, Col. Nueva Industrial Vallejo, C.P. 07738, México, D.F. Tel. 57 29 60 00, ext. 56571.

Editor responsable: *Grigori Sidorov, RFC SIGR651028L69*

Research in Computing Science is published by the Center for Computing Research of IPN. **Volume 93**, May 2015. Printing 500. The authors are responsible for the contents of their articles. All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without prior permission of Centre for Computing Research. Printed in Mexico City, in the IPN Graphic Workshop – Publication Office.

Volume 93

Gestión inteligente de información y reconocimiento de patrones

Obdulia Pichardo Lagunas (ed.)



Instituto Politécnico Nacional
"La Técnica al Servicio de la Patria"



Instituto Politécnico Nacional, Centro de Investigación en Computación
México 2015

ISSN: 1870-4069

Copyright © Instituto Politécnico Nacional 2015

Instituto Politécnico Nacional (IPN)
Centro de Investigación en Computación (CIC)
Av. Juan de Dios Bátiz s/n esq. M. Othón de Mendizábal
Unidad Profesional “Adolfo López Mateos”, Zacatenco
07738, México D.F., México

<http://www.rcs.cic.ipn.mx>

<http://www.ipn.mx>

<http://www.cic.ipn.mx>

The editors and the publisher of this journal have made their best effort in preparing this special issue, but make no warranty of any kind, expressed or implied, with regard to the information contained in this volume.

All rights reserved. No part of this publication may be reproduced, stored on a retrieval system or transmitted, in any form or by any means, including electronic, mechanical, photocopying, recording, or otherwise, without prior permission of the Instituto Politécnico Nacional, except for personal or classroom use provided that copies bear the full citation notice provided on the first page of each paper.

Indexed in LATINDEX and Periodica / Indexada en LATINDEX y Periódica

Printing: 500 / Tiraje: 500

Printed in Mexico / Impreso en México

Editorial

El propósito de este volumen de la revista RCS es reflejar las nuevas direcciones de investigación y aplicaciones de los métodos de la Inteligencia Artificial en algunas áreas específicas.

Los artículos de este volumen fueron seleccionados con base en un estricto proceso de revisión efectuada por los miembros del Comité de revisión, tomando en cuenta la originalidad, aportación y calidad técnica de los mismos. Cada artículo fue revisado por lo menos por dos miembros del Comité de revisión del volumen.

Este volumen contiene 15 artículos relacionados con varios aspectos del desarrollo de los métodos de Inteligencia Artificial y ejemplos de sus aplicaciones a varias tareas tales como:

- Detección de emociones en voz,
- Clasificación automática de documentos,
- Manejo de documentos en Internet y bibliotecas digitales,
- Aplicaciones de minería de datos,
- Evaluación de belleza artificial, entre otras.

Este volumen puede ser interesante para los investigadores y estudiantes de las ciencias de la computación, especialmente en áreas relacionadas con la inteligencia artificial y su aplicación a los diferentes ámbitos de la vida cotidiana; así como, para el público en general interesado en estos fascinantes temas.

En este número especial de la revista RCS, a nombre de la comunidad académica del Centro de Investigación e Innovación en Tecnologías de la Información y Comunicación (INFOTEC) y de la SMIA expresamos nuestro agradecimiento al Dr. Sergio Carrera Riva Palacio, Director Ejecutivo, y Dr. Juan Carlos Téllez Mosqueda, Director Adjunto de Innovación y Conocimiento, por apoyar de manera ingente la investigación y el desarrollo de la ciencia y la tecnología, sustentado todo ello en la responsabilidad y el compromiso social.

El proceso de revisión y selección de artículos se llevó a cabo usando el sistema libremente disponible EasyChair, www.EasyChair.org.

Obdulia Pichardo Lagunas
Mayo 2015

Table of Contents

	Page
Integración de optimización evolutiva para el reconocimiento de emociones en voz.....	9
<i>Luis-Alberto Pérez-Gaspar, Santiago-Omar Caballero-Morales, Felipe Trujillo-Romero</i>	
Predicción del CTR de los anuncios de Internet usando redes orgánicas artificiales	23
<i>Luis Miralles Pechuán, Hiram Ponce</i>	
Análisis del desempeño del algoritmo genético en la clasificación automática de documentos	33
<i>Juan Manuel Zárate Sánchez, Hilarión Muñoz Contreras, María Antonieta Abud Figueroa</i>	
Sistema multiagente integrador de bibliotecas digitales	45
<i>Christian Sánchez-Sánchez, Héctor Jiménez-Salazar, Carlos Rodríguez-Lucatero, Esaú Villatoro-Tello, Gabriela Ramírez-de-la-Rosa</i>	
Detección de patrones de movilidad en datos de localización aplicando modelos de tópicos	57
<i>Manuel Arturo Suárez-Améndola, Salvador Ruiz-Correa, Rogelio Hasimoto-Beltrán</i>	
Análisis de deserción escolar con minería de datos	71
<i>José Luis Aguirre Mendiola, Rosa María Valdovinos Rosas, Juan Alberto Antonio Velazquez, Roberto Alejo Eleuterio, José Raymundo Marcial Romero</i>	
Reconocimiento de dígitos escritos a mano mediante métodos de tratamiento de imagen y modelos de clasificación	83
<i>Luis Miralles Pechuán, Dafne Rosso Pelayo, Jorge Brieva</i>	
Aproximación de funciones con EPWavenets	95
<i>Alejandro Romero Herrera, Oscar Herrera Alcántara, Victor Manuel Landassuri Moreno</i>	
Hacia la construcción de una herramienta para la creación de bitácoras para Android	111
<i>Bárbara Cervantes, Miguel González-Mendoza, Raúl Monroy, Christian Eduardo Galdámez Blanco, Eduardo Ismael García Pérez</i>	
Belleza artificial: evolucionando partes del rostro.....	121
<i>Ricardo Solano Monje, Nayeli Joaquinita Meléndez Acosta, Sergio Juárez Vázquez, Homero Vladimir Ríos Figueroa</i>	
Eigenespacios de belleza paramétricos como máquina calificadoras.....	133
<i>Ricardo Solano Monje, Nayeli Joaquinita Meléndez Acosta, Sergio Juárez Vázquez, Homero Vladimir Ríos Figueroa</i>	

An Estimation of Distribution Algorithm based Approach for the Order Batching Problem.....	141
<i>Ricardo Pérez-Rodríguez, Arturo Hernández-Aguirre</i>	
Mecanismo de transformación de diagramas UML de casos de uso a código WSCDL.....	151
<i>Isaac Machorro Cano, Daniel Eduardo Díaz Navarro, Giner Alor Hernández, Mónica Guadalupe Segura Ozuna, Lisbeth Rodríguez Mazahua</i>	
Factores predisponentes en relajación residual neuromuscular	163
<i>María Dolores Torres Soto, Aurora Torres Soto, María de la Luz Torres Soto, Leticia Bermudez Rosales, Eunice E. Ponce de León Sentí</i>	
Metamodelo de aprendizaje estratégico (MAE): Arquitectura de la capa de infraestructura, solución basada en la Cloud Computing	175
<i>Rafaela Blanca Silva-López, Iris Iddaly Méndez-Gurrola, Oscar Herrera Alcántara</i>	

Integración de optimización evolutiva para el reconocimiento de emociones en voz

Luis-Alberto Pérez-Gaspar, Santiago-Omar Caballero-Morales, Felipe Trujillo-Romero

Universidad Tecnológica de la Mixteca, Huajuapán de León, Oaxaca, México

`luis_335450@hotmail.com`, `scaballero@mixteco.utm.mx`,
`ftrujillo@mixteco.utm.mx`

Resumen. En este artículo se presenta el desarrollo de un sistema de reconocimiento de emociones basado en la voz. Se consideraron las siguientes emociones básicas: Enojo, Felicidad, Neutro y Tristeza. Para este propósito una base de datos de voz emocional fue creada con ocho usuarios Mexicanos con 640 frases (8 usuarios \times 4 emociones \times 20 frases por emoción). Los Modelos Ocultos de Markov (Hidden Markov Models, HMMs) fueron usados para construir el sistema de reconocimiento. Basado en el concepto de modelado acústico de vocales específicas emotivas un total de 20 fonemas de vocales (5 vocales \times 4 emociones) y 22 fonemas de consonantes fueron considerados para el entrenamiento de los HMMs. Un Algoritmo Genético (Genetic Algorithm, GA) fue integrado dentro del proceso de reconocimiento para encontrar la arquitectura más adecuada para el HMM para cada vocal específica emotiva. Una tasa de reconocimiento total aproximada del 90.00% fue conseguida con el reconocedor de voz construido con los HMMs optimizados.

Palabras clave: Reconocimiento de Emociones por Voz, Algoritmos Genéticos, Modelos Ocultos de Markov.

1. Introducción

Avances tecnológicos recientes en el campo de la Robótica se han centrado en el desarrollo de sistemas capaces de identificar un estado emocional de forma precisa. Dentro de este contexto la computación emocional está siendo integrada en los Robots con el propósito de establecer una interacción más natural y fluida con los humanos. Este objetivo es particularmente importante para el desarrollo de tecnología de asistencia, diagnóstico psiquiátrico y detección de mentiras [12].

Investigaciones sobre el reconocimiento de emociones en la voz han sido reportadas en la literatura. Por ejemplo en [11] una comparativa de desempeño de las técnicas de discriminante lineal de Fisher, Máquinas de Soporte Vectorial (Support Vector Machine, SVM), Análisis de Componente Principal (Principal Component Analysis, PCA) y Redes Neuronales Artificiales (Artificial Neural Networks, ANN) fue presentada para el reconocimiento de emociones.

El discriminante lineal de Fisher y PCA fueron usados para la extracción de características mientras que SVM y ANNs fueron usados para la clasificación. Seis emociones (Enojo, Felicidad, Tristeza, Sorpresa, Miedo y Disgusto) fueron consideradas. La tasa de reconocimiento para el sistema Fisher+SVM fue del 50.16% mientras que para el sistema PCA+ANN fue del 39.16%. El trabajo reportó una confusión significativa entre Felicidad y Sorpresa, al igual que la necesidad de mejorar la tasa de reconocimiento para Miedo y Disgusto.

Otro sistema de reconocimiento basado en la voz fue implementado para el Robot MEXI (Machine with Emotionally eXtended Intelligence) [2]. Este sistema permitía un diálogo natural con los usuarios humanos a través de un sistema de síntesis de voz con acento emocional. Cinco emociones (Enojo, Miedo, Tristeza, Felicidad y Neutro) fueron consideradas y la clasificación fue llevada a cabo con un sistema basado en lógica difusa llamado PROSBER. Las tasas de reconocimiento obtenidas fueron aproximadamente del 84.00% para un sistema dependiente de usuario y del 60.00% para un sistema independiente de usuario.

En [16] un reconocimiento multimodal de emociones para Enojo, Felicidad, Sorpresa, Miedo, Tristeza y Neutro fue desarrollado usando FAPS (Facial Animation Parameters) y la técnica de Lipschitz para características acústicas. Modelos Ocultos de Markov Triples (Tripled Hidden Markov Models, THMMs) fueron implementados para realizar la sincronización del audio con las secuencias de patrones visuales y su clasificación. Para el sistema de voz una tasa de reconocimiento de 81.44% fue obtenida mientras que para el sistema visual la tasa fue de 87.40%. Sin embargo para el sistema multimodal (voz+visión) la tasa de reconocimiento fue alrededor de 93.30%.

Finalmente en [6] una SVM multi-clase fue desarrollada para el reconocimiento de cinco emociones (Enojo, Miedo, Felicidad, Neutro y Tristeza). Los Coeficientes Cepstrales en las Frecuencias de Mel (Mel-Frequency Cepstral Coefficients, MFCCs), Histogramas de Periodicidad y Patrones de Fluctuación fueron usados para la extracción de características. Experimentos realizados con la base de datos de voz emocional danesa DES (Danish Emotion Speech) presentaron las siguientes tasas de reconocimiento: 64.77% con función Kernel Lineal, 78.41% con función Polinomial, 79.55% con función RBF y 78.41% con función Sigmoide. Una confusión significativa fue observada entre Felicidad y Enojo.

En este artículo se aborda el reconocimiento de emociones considerando el Español Mexicano. Para esto se desarrolló una base de datos de voz emocional con usuarios Mexicanos. Para la tarea de reconocimiento se utilizó la técnica de modelado acústico de vocales específicas emotivas con HMMs [5]. Mientras que en otros trabajos una estructura HMM estándar es considerada para el reconocimiento de emociones en la voz [5,8,17] en este trabajo un Algoritmo Genético (GA) fue diseñado para encontrar la estructura más adecuada para los HMMs de cada vocal específica emotiva. Los resultados obtenidos mostraron que las características acústicas asociadas a las vocales de cada emoción requieren estructuras específicas de HMMs lo cual puede mejorar su reconocimiento.

La estructura del presente trabajo es la siguiente: en la Sección 2 los detalles del sistema de reconocimiento basado en voz son presentados. Estos detalles

incluyen la creación del corpus de voz emocional y el transcriptor fonético asociado para el entrenamiento supervisado de los HMMs. Después en la Sección 3 se presenta el diseño del GA para encontrar la estructura más adecuada de los HMMs para el modelado acústico. Los resultados del sistema HMM con la optimización del GA son presentados y discutidos en la Sección 4. Finalmente en la Sección 5 se presentan las conclusiones y el trabajo a futuro.

2. Sistema de reconocimiento emocional por voz

Para el desarrollo de un sistema de reconocimiento de emociones es importante contar previamente con una base de datos apropiada para el entrenamiento (modelado) del mismo. Para el presente trabajo una base de datos de voz (corpus) emocional fue requerida. Aunque existen corpora de voz de este tipo para propósitos de investigación la mayoría de los mismos se encuentran en lenguajes extranjeros (por ejemplo, Inglés [3,9,15] y Alemán [1,15]). Estos recursos no pueden ser fácilmente adaptados para otros lenguajes porque hay diferencias fonéticas entre ellos.

Dada esta situación fue necesaria la creación de un corpus de voz emocional Mexicano. Las siguientes condiciones fueron consideradas para el desarrollo de este recurso [5,13]:

- estímulo textual de diferentes longitudes para cada emoción;
- significancia semántica de los estímulos textuales;
- deben haber suficientes ocurrencias de las vocales específicas emotivas y consonantes en el texto de estímulo.

Los voluntarios para la base de datos emocional estuvieron dentro del grupo de edades de los 16 a los 53 años y no fueron actores profesionales. Para tener una pronunciación estándar Mexicana estos voluntarios fueron reclutados de las regiones Este y Sur-Oeste de México. Un total de cinco mujeres y tres hombres fueron considerados para el corpus de voz emocional.

2.1. Base de datos de voz

Previo a la grabación de las muestras de voz se diseñó el estímulo textual para cada emoción. Esto fue importante para tener muestras de voz con la entonación emocional apropiada. Debido a que se ha encontrado en la literatura que las propiedades espectrales de los sonidos de las vocales son un indicador confiable de las emociones en la voz [9,10] éstas pueden ser usadas para el reconocimiento de emociones si se les considera fonéticamente independientes en la creación de un sistema de reconocimiento de voz estándar [5]. De esta forma es considerado que una vocal “a” expresada con Enojo es diferente de una “a” expresada con Tristeza o Felicidad. Esto permite el modelado acústico de vocales específicas emotivas [5].

Para este trabajo las siguientes emociones fueron consideradas: Enojo, Felicidad, Neutro y Tristeza [5,18,20]. El texto de estímulo para Enojo, Felicidad y

Tristeza consistió de frases que fueron concebidas en el contexto de situaciones de la vida cotidiana. Para Neutro las frases fueron consideradas de cultura general. Se diseñaron 20 frases para cada emoción y algunos ejemplos son presentados en la Tabla 1.

Tabla 1. Muestra de Frases de Estímulo Diseñadas para cada Emoción.

Frases para Enojo	
1	¡Yo no te voy a estar soportando!
2	¡Ya me tienes hartos, ya deja de hablar!
3	...
Frases para Felicidad	
1	¡Me gané un viaje todo pagado a Florida!
2	¡Me compré un billete de lotería y gané!
3	...
Frases para Neutro	
1	El graznido de un pato no hace eco
2	La araña Sidney es la más venenosa y puede matar a un humano en 15 minutos
3	...
Frases para Tristeza	
1	Mi mejor amigo acaba de fallecer ayer
2	Me haces mucha falta te extraño
3	...

Para asegurar el modelado acústico apropiado de las vocales un mínimo de seis ocurrencias fue considerado. En la Tabla 2 se presenta el número de muestras por vocales para cada grupo de frases emocionales. Nótese que el mínimo es de 19 muestras (“u” con Tristeza) lo cual es mayor que el número mínimo considerado de seis ocurrencias.

Tabla 2. Número de Vocales por Grupo de Frases de Estímulo.

Vocal	Enojo	Felicidad	Neutro	Tristeza
a	65	86	92	83
e	83	94	115	86
i	38	46	60	58
o	54	54	74	65
u	23	28	35	19

Las frases emocionales fueron grabadas en un salón a puerta cerrada con la herramienta Wavesurfer [4] en formato .WAV con una frecuencia de muestreo de 48000 Hz. La distancia entre el micrófono (micrófono interno de una compu-

tadora tipo laptop) y el usuario fue de alrededor de 60 cm. A cada voluntario se le pidió pronunciar cada una de las 20 frases por emoción llegando a un total de 80 muestras de voz por voluntario (80 frases \times 8 usuarios = 640 frases).

2.2. Etiquetado fonético para el modelado acústico

Después de que las muestras de voz fueron grabadas, los archivos de audio fueron etiquetados a nivel palabra con Wavesurfer como se presenta en la Figura 1. Para identificar las palabras y (subsecuentemente) los fonemas de las vocales las cuales fueron pronunciadas con una emoción en particular un identificador fue añadido a la palabras y a las etiquetas fonéticas. Para cada emoción el identificador para las palabras fue *_E* para Enojo, *_F* para Felicidad, *_N* para Neutro y *_T* para Tristeza. Para las vocales (a nivel fonético) los identificadores fueron *_e*, *_f*, *_n* y *_t* respectivamente [5].

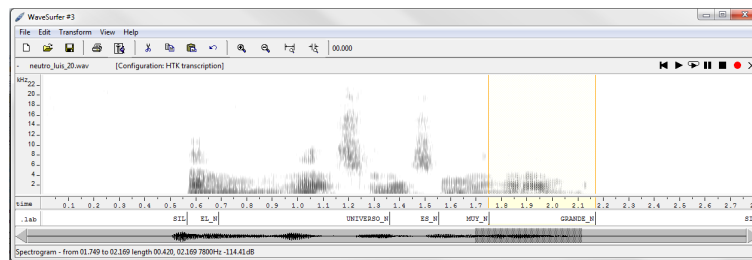


Fig. 1. Etiquetado a Nivel Palabra con Wavesurfer.

Una vez que el proceso de etiquetado a nivel palabra fue terminado se llevó a cabo el etiquetado a nivel fonético. En el Español Mexicano existen 27 fonemas (22 consonantes + 5 vocales) [7]. Debido a que un grupo de vocales fue considerado para cada emoción un total de 20 vocales (5 vocales \times 4 emociones) fueron integradas dentro del repertorio fonético para el Español Mexicano lo cual llevó a un total de 42 fonemas (22 consonantes + 20 vocales).

Para obtener la secuencia de los fonemas para cada palabra un transcriptor fonético basado en TranscribEmex [14] fue desarrollado. El transcriptor fonético consideró aproximadamente 60 reglas gramaticales y acústicas para las diferentes combinaciones de vocales y consonantes dentro de una palabra. Algunas de las reglas se presentan a continuación:

- Si la consonante “q” (fonema /k/) aparece antes de la vocal “u” y la vocal “e” o “i” sigue a ésta entonces la vocal “u” no tiene sonido y el fonema asociado (por ejemplo: /u_e/) no se incluye en la transcripción (por ejemplo: “QUE” \rightarrow /k/ /e_n/, “QUIEN” \rightarrow /k/ /i_n/ /e_n/ /n/ si las palabras fueron pronunciadas con la emoción Neutro).

- Si la consonante “n” aparece al principio de la palabra el fonema asociado en la transcripción es /n/. Sin embargo si la consonante aparece al final el fonema que representa su sonido es /_N/.
- Si la consonante “d” aparece al principio de una palabra, o si una vocal o la consonante “r” le sigue, entonces el fonema que representa su sonido es /d/. Sin embargo, si “d” aparece al final de la palabra, o después de una vocal, el sonido asociado es mejor descrito con el fonema /_D/ (por ejemplo: “DRAGON” → /d/ /r(/ /a_e/ /g/ /o_e/ /_N/, “DIGNIDAD” → /d/ /i_t/ /_G/ /n/ /i_t/ /d/ /a_t/ /_D/ si las palabras fueron expresadas con Enojo y Tristeza respectivamente).
- Si la consonante “g” aparece al final de una palabra su sonido es representado con /_G/. Sin embargo si las consonantes “r” o “l”, o las vocales “a”, “o”, o “u” aparecen después de la consonante “g”, entonces el sonido es mejor descrito con el fonema /g/. Cuando la vocal “e” o “i” aparece después de la “g” entonces el fonema correcto es /x/ (por ejemplo: GLOBO → /g/ /l/ /o_e/ /b/ /o_e/, GRITAR → /g/ /r(/ /i_e/ /t/ /a_e/ /_R/, GENIO → /x/ /e_e/ /n/ /i_e/ /o_e/, GITANA → /x/ /i_e/ /t/ /a_e/ /n/ /a_e/ si las palabras fueron pronunciadas con Enojo).

2.3. Modelo de lenguaje

El modelo de lenguaje es un elemento importante de cualquier sistema de reconocimiento de voz ya que contiene información estadística de las secuencias válidas de palabras en un lenguaje. El lenguaje Español contiene varias estructuras acerca de las secuencias de palabras que forman oraciones coherentes. Es importante mencionar que cualquier oración o frase puede ser pronunciada con cualquier emoción. De esta forma las estructuras gramaticales en un lenguaje aplican a cualquier emoción.

El modelado específico de emociones implica la independencia de vocales pero no de las palabras del vocabulario. A pesar de que las palabras habladas con una emoción dada tienen un identificador (*_E*, *_F*, *_N* o *_T*) estas palabras existen para todas las emociones. De esta forma el modelo de lenguaje fue integrado por el conjunto completo de 80 frases considerando que cada una de ellas puede ser expresada con todas las emociones. Esto llevó a un total de 80×4 emociones = 320 frases para la estimación del modelo de lenguaje para el sistema de reconocimiento de voz. Esto también fue requerido para evitar un sesgo o influencia en el reconocimiento del estado emocional por parte del modelo de lenguaje.

Es importante mencionar que el reconocimiento de emociones es estimado contando el número de vocales dentro de las palabras reconocidas. El identificador (*_e*, *_f*, *_n*, *_t*) con el mayor número de vocales define la emoción dominante.

3. Optimización de HMMs con algoritmos genéticos

Todos los sonidos (fonemas) identificados en los archivos de audio con sus transcripciones fonéticas deben ser modelados para poder ser reconocidos. Entre

las técnicas usadas para modelado fonético los HMMs han sido ampliamente usados [19]. En la Figura 2(a) se presenta la estructura Bakis que es la más común para este propósito [19]. Sin embargo para el modelado acústico de vocales específicas emotivas otras estructuras pueden ser más adecuadas. La Figura 2(b) y la Figura 2(c) presentan estructuras HMM alternativas para el modelado acústico de fonemas. El problema de identificar la estructura HMM apropiada para cada vocal específica emotiva puede ser resuelto con un Algoritmo Genético (GA).

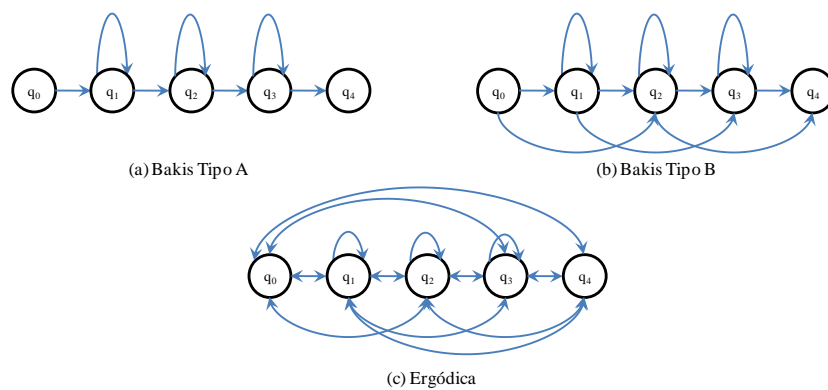


Fig. 2. Estructuras HMM para Modelado Acústico de los Fonemas de las Vocales.

Un GA es una técnica de optimización heurística que se basa en el proceso natural de sobrevivencia y adaptación de los individuos más aptos en una población. Estos individuos al sobrevivir son más probables a reproducirse, convirtiéndose en “padres” de nuevas generaciones de “hijos” que heredarán las características de los hicieron más aptos para sobrevivir y adaptarse al entorno. Estas características se van mejorando en cada ciclo de reproducción de manera generacional. Dentro del contexto de los GA los individuos (“padres” e “hijos”) representan posibles soluciones a un problema combinatorio.

El diagrama general de operación y módulos principales del GA para el presente trabajo se presentan en la Figura 3. El cromosoma para la optimización de la estructura de los HMMs consistió de 20 (2-bit) genes (5 vocales \times 4 emociones) en donde cada gen contiene el tipo de estructura de HMM para la vocal/emoción asociada. Solamente las estructuras de las vocales específicas emotivas fueron consideradas para optimización. Los modelos HMM para las consonantes tuvieron una estructura estándar “Bakis Tipo A”. El valor de la aptitud de los individuos (función objetivo) fue medido como la tasa de clasificación obtenida con el conjunto completo de HMMs.

Para encontrar las estructuras de HMMs más adecuadas cada conjunto de frases fue dividido en: (a) frases de entrenamiento y (b) frases para optimización (evaluación de aptitud). El conjunto de entrenamiento consistió de las últimas 8

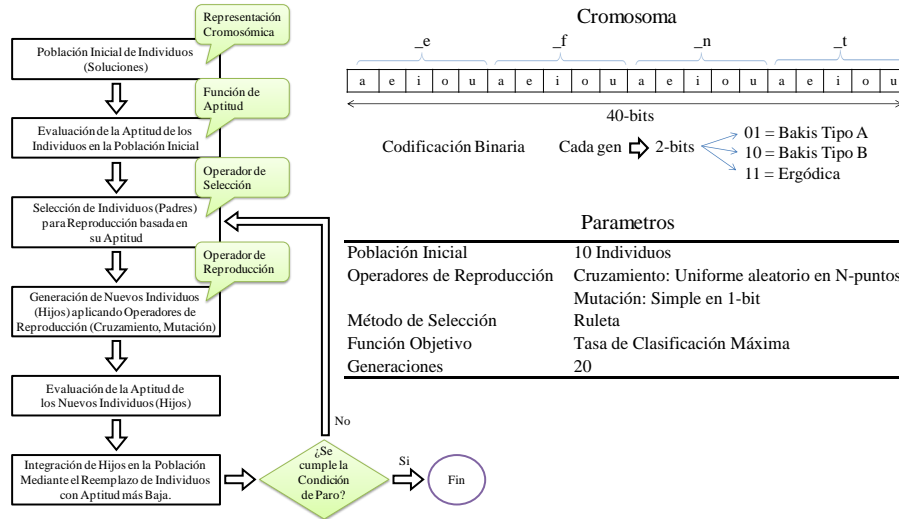


Fig. 3. Diagrama, Cromosoma y Parámetros de Configuración del Algoritmo Genético.

frases de cada conjunto emocional (frases 13 a 20) y las frases de optimización consistieron de las seis frases intermedias (frases 7 a 12). Las primeras seis frases (frases 1 a 6) fueron consideradas para la evaluación preliminar del GA.

En la Figura 4 se presenta el vector fila resultante con las estructuras de HMMs para cada vocal específica emotiva. También se presenta el desempeño preliminar del reconocimiento de emociones con estas estructuras sobre las frases de evaluación para todos los usuarios. Este desempeño es comparado con el de un reconocedor en donde todos los HMMs tienen la misma estructura estándar (Bakis Tipo A). Como se presenta, el conjunto de HMMs encontrados por el GA obtuvieron una ganancia significativa del 5.20% (75.00% - 80.20%) sobre las frases de evaluación. En este conjunto se observa una combinación de todas las estructuras consideradas (Bakis Tipo A, Bakis Tipo B, Ergódica) en donde la estructura Bakis Tipo B tiene más presencia.

4. Resultados

Para la evaluación final del enfoque evolutivo con GA para el reconocimiento de emociones basado en voz dos esquemas fueron considerados:

- Esquema de Prueba A (dependiente de usuario): bajo este esquema 40 frases (10 primeras frases \times 4 emociones) de cada usuario fueron consideradas para entrenamiento de los HMM adicionalmente a las 560 frases (20 frases \times 4 emociones \times 7 usuarios restantes) de los otros usuarios. Finalmente el desempeño del reconocimiento es evaluado con el resto de las 40 frases del hablante en cuestión (10 últimas frases \times 4 emociones).

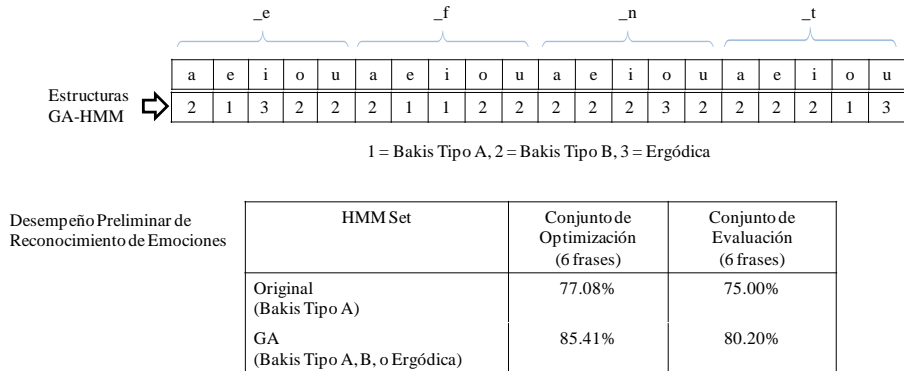


Fig. 4. GA-HMMs para las Vocales Específicas Emotivas y Desempeño Preliminar de Reconocimiento de Emociones.

- Esquema de Prueba B (independiente de usuario): bajo este esquema 40 frases (10 primeras frases \times 4 emociones) de cada usuario fueron consideradas para adaptación de usuario. Los HMMs fueron entrenados solamente con las 560 frases (20 frases \times 4 emociones \times 7 usuarios restantes) de los otros usuarios. Finalmente el desempeño del reconocimiento es evaluado con el resto de las 40 frases del hablante en cuestión (10 últimas frases \times 4 emociones).

La herramienta HTK [19] fue utilizada para el desarrollo del sistema de reconocimiento de voz con HMMs y la implementación de la técnica de adaptación de usuario (Maximum Likelihood Linear Regression, MLLR) bajo el esquema independiente de usuario. Para la codificación de las muestras de voz se utilizó la técnica de vectores de características espectrales MFCC (Mel Frequency Cepstral Coefficients). Para esto se utilizaron 12 MFCCs más los coeficientes de energía, delta y de aceleración [19]. Los desempeños de las estructuras estándar (Bakis Tipo A solamente) y las estructuras GA-HMM (ver Figura 4) fueron evaluados bajo ambos esquemas de prueba y los resultados son presentados en la Tabla 3 y Tabla 4.

Para ambos sistemas (HMMs Estándar y GA-HMMs) el esquema de prueba independiente de usuario presentó un desempeño mayor que el del esquema dependiente de usuario. Para la validación estadística de la mejora obtenida con el GA se hizo uso de la prueba no paramétrica de Wilcoxon de una muestra. Esto dado que los resultados no tienen una distribución normal. La prueba de Wilcoxon puede determinar si la media de un conjunto de datos difiere de un valor en específico (referencia). Para el Esquema de Prueba A (dependiente de usuario) se consideró como valor de referencia el promedio total obtenido con los HMMs Estándar (78.59%). Al analizar el conjunto de datos del Esquema de Prueba A correspondiente al reconocimiento con los GA-HMMs se obtuvo que hay una diferencia significativamente estadística con $p=0.065$ (considerando $p < 0.10$).

Tabla 3. Desempeño del Sistema de Reconocimiento de Emociones por Voz: HMMs Estándar.

Esquema de Prueba A						Esquema de Prueba B					
Usuario	Género	Enojo	Felicidad	Neutro	Tristeza	Usuario	Género	Enojo	Felicidad	Neutro	Tristeza
Lu	M	100.00	50.00	100.00	80.00	Lu	M	100.00	50.00	100.00	100.00
Ta	F	100.00	80.00	100.00	90.00	Ta	F	100.00	70.00	100.00	100.00
Au	F	80.00	85.00	80.00	100.00	Au	F	100.00	100.00	80.00	100.00
Mi	M	70.00	70.00	100.00	85.00	Mi	M	70.00	80.00	100.00	90.00
Me	F	75.00	70.00	90.00	90.00	Me	F	95.00	90.00	100.00	100.00
Je	M	100.00	30.00	75.00	50.00	Je	M	80.00	100.00	70.00	90.00
Li	F	70.00	40.00	20.00	75.00	Li	F	75.00	80.00	75.00	70.00
Ne	F	80.00	100.00	90.00	90.00	Ne	F	90.00	100.00	100.00	80.00
Promedio		84.38	65.63	81.88	82.50	Promedio		88.75	83.75	90.63	91.25
Promedio Total					78.59	Promedio Total					88.59

Tabla 4. Desempeño del Sistema de Reconocimiento de Emociones por Voz: GA-HMMs.

Esquema de Prueba A						Esquema de Prueba B					
Usuario	Género	Enojo	Felicidad	Neutro	Tristeza	Usuario	Género	Enojo	Felicidad	Neutro	Tristeza
Lu	M	100.00	60.00	100.00	90.00	Lu	M	100.00	60.00	100.00	100.00
Ta	F	100.00	90.00	100.00	90.00	Ta	F	100.00	90.00	100.00	90.00
Au	F	80.00	70.00	80.00	100.00	Au	F	100.00	100.00	80.00	100.00
Mi	M	100.00	65.00	100.00	90.00	Mi	M	70.00	60.00	90.00	90.00
Me	F	65.00	90.00	100.00	90.00	Me	F	95.00	100.00	90.00	100.00
Je	M	100.00	20.00	85.00	25.00	Je	M	90.00	100.00	90.00	80.00
Li	F	60.00	45.00	80.00	90.00	Li	F	90.00	60.00	90.00	70.00
Ne	F	80.00	100.00	100.00	90.00	Ne	F	100.00	100.00	100.00	80.00
Promedio		85.63	67.50	93.13	83.13	Promedio		93.13	83.75	92.50	88.75
Promedio Total					82.34	Promedio Total					89.53

Sin embargo bajo el Esquema de Prueba B (independiente de usuario) la mejora obtenida con los GA-HMMs no fue estadísticamente significativa. Considerando como valor de referencia el promedio total obtenido con los HMMs Estándar (88.59 %) la prueba de Wilcoxon determinó que el conjunto de datos correspondiente al reconocimiento con los GA-HMMs no era estadísticamente diferente dado $p=0.147$ ($p > 0.10$). A pesar de que se obtuvieron mejoras para Enojo y Neutro con los GA-HMMs bajo el esquema independiente de usuario no hubo una mejora para Tristeza.

Para ambos sistemas y esquemas de prueba Felicidad fue la emoción con la tasa más baja de reconocimiento. Considerando el uso de la estructura estándar solamente para las vocales específicas emotivas de Tristeza (Estándar-GA-HMMs) bajo el Esquema de Prueba B el desempeño total del sistema se presenta en la Tabla 5. Este desempeño (90.16 %) es marginalmente significativo comparado con el valor de referencia de los HMMs Estándar (88.59 %) al tener $p=0.091$.

Tabla 5. Desempeño del Sistema de Reconocimiento de Emociones por Voz: Estándar-GA-HMMs.

Sistema	Esquema de Prueba B				Promedio
	Enojo	Felicidad	Neutro	Tristeza	
HMMs Estándar	88.75	83.75	90.63	91.25	88.59
GA-HMMs	93.13	83.75	92.50	88.75	89.53
Estándar-GA-HMMs	93.13	83.75	92.50	91.25	90.16

5. Conclusiones

Las estructuras de HMMs estimadas con el GA estadísticamente mejoraron el desempeño del reconocimiento de emociones bajo el esquema de prueba dependiente de usuario (Esquema de Prueba A) de 78.59 % a 82.34 %. A pesar de que el desempeño de reconocimiento total fue mayor bajo el esquema de prueba independiente de usuario (Esquema de Prueba B) y se obtuvo un incremento adicional con el GA (88.59 % a 89.53 %) éste no fue estadísticamente significativo. Un incremento adicional marginalmente significativo (90.16 %) fue obtenido en el Esquema de Prueba B al considerar las estructuras HMM estándar para la emoción de Tristeza.

El trabajo a futuro se enfocará en mejorar el reconocimiento bajo el esquema de prueba independiente de usuario e incrementar el tamaño de la base de datos emocional. De igual manera mejorar el desempeño del GA para obtener incrementos más significativos (p.e., con $p < 0.05$) y contar con más alternativas para el tipo de las estructuras HMM para optimización. También es importante considerar la integración de otras técnicas de codificación para la extracción de características espectrales para hacer más eficiente la detección de la emoción. Finalmente el alcance del enfoque presentado en este trabajo debe evaluarse con

otras bases de datos de voz emocional y hacer una comparativa extensa con otros enfoques presentados en la literatura.

Referencias

1. Alter, K., Rank, E., Kotz, S.A.: Accentuation and emotions - two different systems ? In: Proc. ISCA Workshop Speech and Emotion. vol. 1, pp. 138–142 (2000)
2. Austermann, A., Esau, N., Kleinjohann, L., Kleinjohann, B.: Fuzzy emotion recognition in natural speech dialogue. In: Proc. of the 14th IEEE International Workshop on Robot and Human Interactive Communication (RO-MAN 2005) (2005)
3. Batliner, A., Hacker, C., Steidl, S., Nöth, E., D’Archy, S., Russell, M., Wong, M.: “you stupid tin box” - children interacting with the AIBO robot: A cross-linguistic emotional speech corpus. In: Proc. Language Resources and Evaluation (LREC ’04) (2004)
4. Beskow, J., Sjolander, K.: WaveSurfer. KTH: The Department of Speech, Music and Hearing (2013)
5. Caballero, S.: Recognition of emotions in mexican spanish speech: An approach based on acoustic modelling of emotion-specific vowels. The Scientific World Journal pp. 1–13 (2013)
6. Chavan, V.M., Gohokar, V.V.: Speech emotion recognition by using SVM-classifier. International Journal of Engineering and Advanced Technology (IJEAT) 1(5), 11–15 (2012)
7. Cuétara, J.: Fonética de la Ciudad de México: Aportaciones desde las Tecnologías del Habla. Tesis de Maestría, Universidad Nacional Autónoma de México (UNAM), México. (2004)
8. Fernandez, R., Picard, R.: Modelling drivers’ speech under stress. Speech Communication 40, 145–159 (2003)
9. Lee, C.M., Yildirim, S., Bulut, M., Kazemzadeh, A., Busso, C., Deng, Z., Lee, S., Narayanan, S.: Emotion recognition based on phoneme classes. In: Proc. Int. Conf. Spoken Language Processing (ICSLP ’04). vol. 1, pp. 889–892 (2004)
10. Li, A., Fang, Q., Hu, F., Zheng, L., Wang, H., Dang, J.: Acoustic and articulatory analysis on Mandarin Chinese Vowels in emotional speech. In: Proc. 7th International Symposium on Chinese Spoken Language Processing (ISCSLP), 2010. pp. 38–43 (2010)
11. Lijiang, C., Mao, X., Xue, Y., Cheng, L.: Speech emotion recognition: Features and classification models. Digital Signal Processing 22, 1154–1160 (2012)
12. Lin, Y.-L., Wei, G.: Speech emotion recognition based on HMM and SVM. In: Proc. of the 2005 International Conference on Machine Learning and Cybernetics. vol. 8, pp. 4898–4901 (2005)
13. López, J.M., Cearreta, I., Garay, N., López de Ipiña, K., Beristain, A.: Creación de una base de datos emocional bilingüe y multimodal. In: Proc. of the 7th Spanish Human Computer Interaction Conference, Interaccion 2006. vol. 6, pp. 55–66 (2006)
14. Pineda, L., Villaseñor, L., Cuétara, J., Castellanos, H., Galescu, L., Juárez, J., Llisterri, J., Pérez, P.: The corpus DIMEX100: Transcription and evaluation. Language Resources and Evaluation 44, 347–370 (2010)
15. Schuller, B., Rigoll, G., Lang, M.: Hidden Markov model-based speech emotion recognition. In: Proc. of the International Conference on Multimedia and Expo. pp. 401–404 (2003)

16. Song, M., You, M., Li, N., Chen, C.: A robust multimodal approach for emotion recognition. *Neurocomputing* 71, 1913–1920 (2008)
17. Wagner, J., Vogt, T., André, E.: A systematic comparison of different HMM designs for emotion recognition from acted and spontaneous speech. *Affective Computing and Intelligent Interaction, Series “Lecture Notes in Computer Science”* 4738, 114–125 (2007)
18. Yildirim, S., Bulut, M., Lee, C.M., Kazemzadeh, A., Busso, C., Deng, Z., Lee, S., Narayanan, S.: An acoustic study of emotions expressed in speech. In: *Proc. Int. Conf. Spoken Language Processing (ICSLP '04)*. vol. 1, pp. 2193–2196 (2004)
19. Young, S., Woodland, P.: *The HTK Book (for HTK Version 3.4)*. Cambridge University Engineering Department, UK. (2006)
20. Yu, F., Chang, E., Xu, Y. Q., Shum, H.Y.: Emotion detection from speech to enrich multimedia content. In: *Proc. IEEE Pacific-Rim Conf. Multimedia 2001*. vol. 1, pp. 550–557 (2001)

Predicción del CTR de los anuncios de Internet usando redes orgánicas artificiales

Luis Miralles Pechuán, Hiram Ponce

Facultad de Ingeniería, Universidad Panamericana,
México

{lmiralles,hponce}@up.edu.mx

Resumen. Para que las redes publicitarias aumenten sus ingresos es necesario darle prioridad a los anuncios más rentables. El factor más importante en la rentabilidad de un anuncio es el CTR (Click-through-rate) que es la probabilidad de que un usuario realice un clic en el anuncio de una página web. Para predecir el CTR, hemos entrenado varios modelos de clasificación supervisados y hemos comparado su rendimiento con las redes ROA (Redes Orgánicas Artificiales). La conclusión es que estas redes son una buena solución para predecir el CTR de un anuncio.

Palabras clave: predicción de CTR, redes orgánicas artificiales en publicidad, modelos supervisados de clasificación, redes de publicidad CPC.

1. Introducción

Desde que se mostrara el primer banner en octubre de 1994, la publicidad en Internet ha tenido un crecimiento constante [1]. La infraestructura y la tecnología de la red permiten lanzar campañas con importantes ventajas para los anunciantes que la publicidad tradicional no puede ofrecer.

Una de las principales ventajas es que permite a los anunciantes segmentar los clientes en función de ciertos parámetros como: localización, horario, aficiones o tipo de dispositivo desde el que se accede. Esto se conoce como *microtargeting* y permite lanzar campañas a un pequeño grupo con intereses comunes. Estas campañas resultan muy eficaces [2].

La publicidad en Internet tiene una amplia variedad de tipos de anuncios como son los *pop-up*, los *pop-under* o los *interstitial*¹. Pero los anuncios más utilizados debido a su gran impacto y a que no son intrusivos son los banners, los anuncios de texto y los videos.

¹ Los pop-up son ventanas emergentes, los pop-under son nuevas ventanas del navegador que se abren de forma automática y los anuncios interstitial son animaciones que suelen ocupar toda la página.

El modelo de publicidad en Internet está formado por cuatro roles claramente diferenciables. Los anunciantes, la red publicitaria, los editores y los usuarios. Los anunciantes son las empresas que pagan a la red de publicidad para que se muestre su anuncio en las páginas de los editores. Los editores son las personas que tienen al menos una página web y que perciben una retribución económica por prestar un espacio de sus páginas en el que se muestran los anuncios. Los usuarios son personas que navegan por Internet y que compran un producto cuando les interesa.

La red de publicidad es la encargada de gestionar todo el proceso de publicidad. Entre las principales tareas se encuentran: la detección de fraude [3], la preservación de la privacidad de los roles y la gestión de pagos. Pero sin duda, la tarea más importante consiste en optimizar el rendimiento del modelo publicitario para que los anunciantes tengan campañas exitosas y para que tanto la red de publicidad como los editores obtengan el máximo rendimiento por sus anuncios.

En la publicidad online existe una multitud de modelos de cobro a los anunciantes como son el pago por mostrar un anuncio un cierto número de días, el pago por enlaces que apunten a la web del anunciante o el pago por canjeo de cupones². Pero los modelos más utilizados con gran diferencia son el CPM, el CPC y el CPA.

En el modelo CPM (*Cost-per-mile*) los anunciantes pagan por cada mil impresiones³ de un anuncio. En el modelo CPC (*Cost-per-click*) se paga cada vez que un usuario hace clic sobre un anuncio y por último, en el modelo CPA (*Cost-per-acquisition*) se paga cuando un usuario que hace clic en el banner de un editor realiza una compra o contrata un servicio.

El modelo más utilizado es el CPC, principalmente porque es el más sencillo de entender para los anunciantes y porque lo han adoptado las principales empresas del sector⁴. Para optimizar el rendimiento de la red CPC hemos de calcular la rentabilidad de los anuncios en las campañas que lanzan los anunciantes.

Dentro del modelo de cobro CPC existen dos conceptos relacionados con el pago de los anunciantes que son el CPC Máximo y el CTR. El CPC Máximo representa el importe más alto que un anunciante está dispuesto a pagar por un clic. Sin embargo, en la mayoría de los casos se le cobra un precio menor que se denomina CPC Real. El CTR (*Click-through rate*) de un anuncio se define como el ratio entre el número de clics en el anuncio y el número impresiones. Conocer con total seguridad el CTR [4] que tendrá un determinado anuncio en el futuro no es posible, pero si logramos predecir el CTR de los anuncios con un margen de error muy pequeño podremos elegir el anuncio que proporcione mayores beneficios.

Existen pocas publicaciones respecto a los algoritmos utilizados por las redes publicitarias⁵ [5]. Esto es algo muy razonable puesto que si se publicasen estos

² El canjeo de cupones consiste en repartir un código asociado con un editor que al asignarle le otorga una comisión.

³ Una impresión significa que un anuncio se muestra en alguna página.

⁴ Las principales empresas del sector son Google, Yahoo y Microsoft.

⁵ Si bien es cierto que existe un famoso artículo de Tuzhilin sobre el sistema de detección de fraude de Google, no hemos encontrado documentación sobre los algoritmos que utilizan las redes de publicidad.

algoritmos las empresas perderían la ventaja competitiva. Ya que cualquiera podría copiar las investigaciones que conllevan tantos años y tantos recursos económicos.

Por otra parte, a las personas con intención de cometer fraude se les facilita vulnerar las redes publicitarias.

Para calcular el CTR utilizaremos modelos supervisados de tipo clasificación. La creación de modelos supervisados es una parte importante del *Data Science*. El *Data Science* consiste en obtener información de utilidad para las organizaciones a partir de datos recolectados. Las técnicas que se aplican provienen de ciencias como lógica, estadística, probabilidad, diseño de modelos o reconocimiento de patrones.

Se han desarrollado numerosas herramientas para simplificar y automatizar estos procesos. Entre las herramientas de software libre más completas y con mejores resultados se encuentra R Studio. Utilizaremos esta herramienta para predecir resultados con diversos métodos como: Splines de regresión multivariante de adaptación, Centroides disminuido más cercano o Análisis discriminante lineal, y posteriormente utilizaremos las ROA (Redes Orgánicas Artificiales) para contrastar los resultados.

Primeramente, entrenaremos las redes ROA y el reto de los modelos con un dataset. Una vez entrenados, probaremos la eficacia de estos con un conjunto de entradas y posteriormente evaluaremos los modelos.

2. Redes de hidrocarburos artificiales y sus aplicaciones en la predicción CTR

En esta sección explicamos brevemente la técnica de aprendizaje automático llamada redes de hidrocarburos artificiales (*artificial hydrocarbon networks, AHN*) y posteriormente exponemos el modelo de predicción CTR que proponemos utilizando este método.

2.1. Redes de hidrocarburos artificiales

La técnica de redes orgánicas artificiales (*artificial organic networks, AON*) es una clase de algoritmos inspirados en los compuestos de química orgánica, la cual permite el empaquetamiento de información (patrones de datos) en módulos llamados moléculas. Además, esta técnica define mecanismos similares a los compuestos químicos orgánicos (heurísticas) que generan estructuras organizadas y óptimas en términos de la energía química. Finalmente, las ROA preservan algunas características químicas como: modularidad, herencia, organización y estabilidad estructural [6].

Con ayuda del método explicado anteriormente, la técnica de redes de hidrocarburos artificiales es una clase de algoritmos de aprendizaje supervisado basados en las ROA. De hecho, el algoritmo está inspirado en los hidrocarburos. De manera similar a los hidrocarburos químicos, las redes de hidrocarburos artificiales únicamente utilizan dos unidades: átomos de hidrógeno y de carbono que pueden ser relacionadas como máximo con uno o cuatro átomos, respectivamente. La unión de estos átomos en unidades pequeñas se conoce como moléculas y la unión de estas

últimas se conocen como compuestos, los cuales incluyen relaciones no lineales entre moléculas. Este tipo de redes son adecuadas para problemas de modelado, así como de sistemas de predicción y búsqueda de patrones en datos desconocidos e inciertos. Algunos de los ejemplos de aplicación de esta técnica pueden encontrarse en la bibliografía [7], [8].

2.2. Modelo de predicción CTR basado en las redes de hidrocarburos artificiales

Nosotros proponemos el uso de redes de hidrocarburos artificiales como un método de aprendizaje supervisado para la construcción de un modelo de predicción CTR. Para esta aplicación, decidimos el uso de redes de hidrocarburos artificiales formadas de un compuesto lineal y saturado que reciba múltiples entradas (categorías de un conjunto de datos especificado) y genere una salida (el valor CTR). La Figura 1 muestra nuestra propuesta del modelo de predicción CTR usando AHN.

Como se muestra en la Figura 1, el modelo de predicción CTR es entrenado usando una base de datos con información del tipo atributo-valor. El proceso de entrenamiento se compone del uso del algoritmo de redes de hidrocarburos artificiales multi-categorías. Después, se puede llevar a cabo una petición basada en el mismo tipo de atributos descritos en la base de datos. Finalmente, el modelo de redes de hidrocarburos artificiales genera el valor respectivo de CTR.

En la siguiente sección, probamos nuestro modelo de predicción CTR basado en las redes de hidrocarburos artificiales utilizando un conjunto de datos de publicidad con su valor CTR esperado.

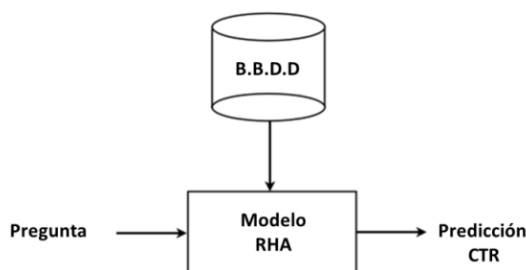


Fig. 1. Nuestra propuesta del modelo de predicción CTR basado en las redes de hidrocarburos artificiales.

2.3. Predicción del CTR del resto de algoritmos mediante Caret de R Studio

R Studio es un entorno de programación orientado al cálculo científico. Esta plataforma permite a los usuarios crear paquetes que implementan un conjunto de funciones o métodos matemáticos. Caret es un paquete diseñado para simplificar el proceso de construcción y evaluación de modelos. Caret dispone de un total de 180

métodos para la construcción de modelos predictivos de los cuales 138 son de clasificación y 42 de regresión [9].

Esta herramienta dispone de métodos que automatizan la selección de variables. Esto permite construir modelos más precisos, menos complejos y en menor tiempo. Los métodos para la selección de variables descartan aquellas características que no aportan información valiosa o que son redundantes. Para este problema utilizaremos un método llamado *Recursive Feature Elimination* [10]. Este método lo aplicamos a todos los modelos menos para el ROA, puesto que este modelo integra un método alternativo llamado PCA (Análisis por componentes principales).

El método RFE es de tipo *Wrapper*, es decir, tiene como entradas una combinación de predictores que utiliza para construir modelos. Y como salida una métrica que evalúa la precisión del modelo mediante el método *random forest*. Esto sirve para ir probando combinaciones de variables hasta lograr la mejor combinación.

La principal ventaja de Caret es la simplicidad del código para construir modelos con distintos métodos y con distintas configuraciones de parámetros. Así como la implementación de una serie de métricas para seleccionar automáticamente la mejor configuración de parámetros de varios modelos que se generan con un método. Para medir la precisión del modelo se suelen utilizar métricas como el ROC, el AUC, la *Accuracy* o el error RMSE.

Cada método tiene un número de parámetros determinado que suele estar entre cero y tres. Por ejemplo, las redes neuronales tienen dos parámetros que son el número de nodos y el nivel de aprendizaje. Caret hace un barrido de posibles valores de los parámetros pero también permite configurar esos parámetros por el programador [11].

Para evaluar los métodos se suelen hacer varios data sets y para ello se utiliza el CV (*Cross-validation*). El método CV sirve para crear distintos training set de forma que podemos construir modelos con el mismo método pero con muestras distintas. En lugar de tomar la precisión de un solo modelo calcula el promedio de todos y de forma que los resultados son más fiables.

Por último, R Studio cuenta con un conjunto de librerías que permite crear gráficos muy ilustrativos y con gran calidad. Esto permite visualizar los resultados obtenidos de una forma mucho más comprensible.

3. Metodología para la predicción de CTR mediante métodos supervisados

Para resolver el problema que se plantea hemos utilizado un dataset proporcionado por la página web: Kaggle.com⁶. Este dataset recoge algunos parámetros de las visitas de los usuarios a la web www.criteo.com durante un periodo de siete días. Sobre estas visitas, se ha reducido en mayor medida el número de muestras en la que los usuarios no hicieron clic en el anuncio que los que sí lo hicieron.

⁶ Esta web se llama Kaggle.com y es famosa por que organiza muchos concursos a nivel mundial.

Las visitas vienen representadas en una tabla donde cada fila representa la visita de un usuario y cada columna representa una característica de un usuario o de la página web. Por ejemplo, en la primera columna de la tabla se representa con “0” o con “1” si el usuario hizo clic en el anuncio. La tabla tiene trece columnas con valores de tipo entero y 26 valores de tipo string que representan ciertas categorías. Los valores de las categorías han sido codificados mediante un hash a un valor de 32 bits para garantizar la privacidad. Las filas están ordenadas cronológicamente y cuando el valor de cierto parámetro se desconoce, simplemente se deja un espacio en blanco.

Para realizar el test de prueba utilizaremos un fichero con el mismo formato que la tabla de entrenamiento pero sin la columna que indica si el usuario hizo clic. Para evaluar el rendimiento de las ROA en la predicción del CTR utilizaremos dos métricas. La primera métrica está basada en el logaritmo de la función de probabilidad para una distribución aleatoria de Bernoulli y la segunda métrica se basa en algo tan sencillo como el porcentaje de aciertos sobre el total de pruebas.

Para entrenar las redes se eligieron un total de 1,000,000 muestras aleatorias, ya que las ROA no soportan excesiva información. De este conjunto de datos, utilizamos el 80% de observaciones aleatorias para el proceso de entrenamiento. El conjunto de datos de entrenamiento se depuró, es decir, se quitaron los datos incompletos. Estas características se eliminaron con ayuda del PCA. Este método selecciona y ordena las columnas más importantes. Posteriormente los datos se estandarizaron usando (1), donde x es una columna del dataset, μ es la media del valor de dicha columna, σ es la desviación estandar de dicha columna, y x_{std} es la columna estandarizada.

Para desarrollar el modelo de predicción CTR la red orgánica se creó un compuesto de 100 moléculas artificiales empleando la técnica de estimación por mínimos cuadrados. Esta red se entrenó con un coeficiente de aprendizaje de 0.5. El coeficiente de aprendizaje es un parámetro de las redes de hidrocarburos artificiales que permite regular la tasa de asimilación o regresión de los datos reales.

$$x_{std} = \frac{x - \mu}{\sigma} \quad (1)$$

4. Resultados obtenidos

En la primera métrica, la probabilidad de que un usuario haga clic se expresa en el rango [0,1]. Cuanto menor sea la estimación de que el usuario no haga clic más se aproximará a cero y cuanto mayor probabilidad estimemos que el usuario hará clic más se aproximará a uno.

Una vez que hayamos establecido los resultados calcularemos el Log Loss del modelo mediante la siguiente fórmula:

$$\text{Log Loss} = -\ln \sum_{i=1} \ln [y_i \log(y_i) + (1-y_i) \log(1-y_i)]$$

El uso del logaritmo permite que una equivocación no tenga un castigo excesivo en el resultado general. Con esta fórmula si decimos que la probabilidad de que haga clic es de un 99% y acertamos nos dará una gran recompensa y si fallamos tendremos una gran penalización.

Los resultados obtenidos con el logaritmo de la función de probabilidad para una distribución aleatoria de Bernoulli han sido muy positivos puesto que hemos obtenido un resultado de Log Loss = 0.6499.

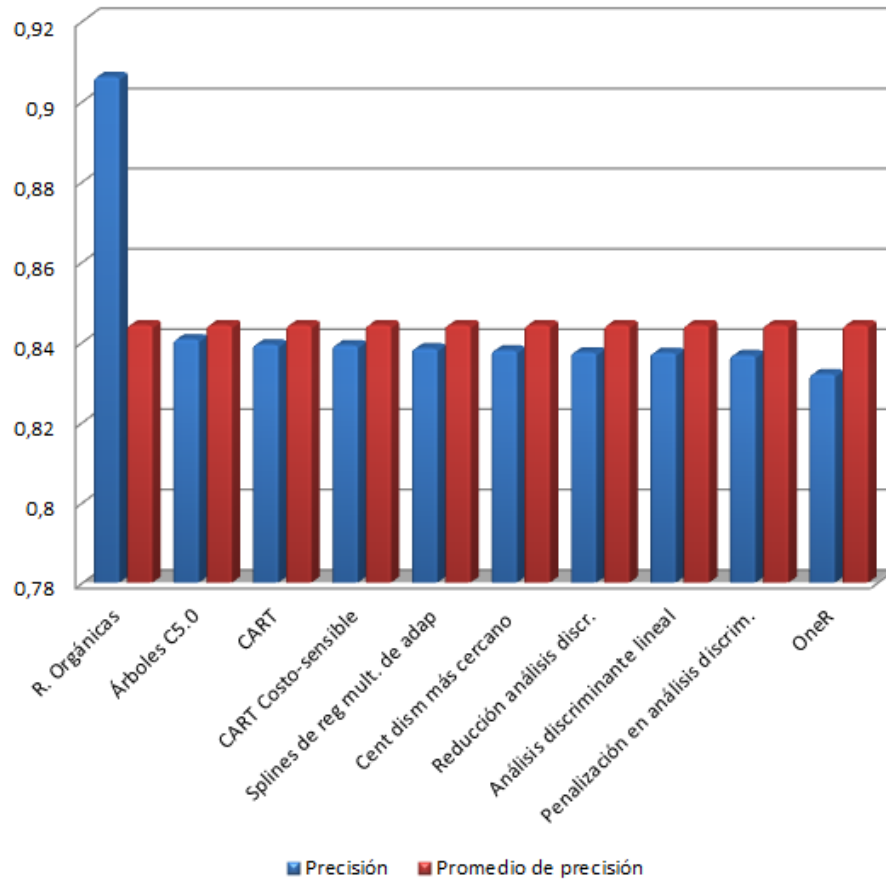


Fig. 2 Gráfica con los resultados de la precisión y el tiempo en horas de las redes.

Vamos a utilizar una segunda métrica más fácil de interpretar. Para medir la eficacia del algoritmo lo expresaremos como el ratio del número de aciertos entre el número de casos totales. Diremos que el usuario hará clic en caso de que la probabilidad estimada sea superior a 0,5 y diremos que no en caso de que sea inferior. El sistema de predicción puede arrojar cuatro posibles resultados:

- Verdadero positivo (VP): Predecimos que el usuario hará clic y el usuario lo hace.
- Verdadero negativo (VN): Predecimos que el usuario no hará clic y si lo hace clic.
- Falso Positivo (FP): Predecimos que el usuario hará clic y no hace clic.
- Falso Negativo (FN): Predecimos que el usuario no hará clic y si hace clic.

Si tenemos en cuenta que $VP + VN + FP + FN = N$. Entonces la tasa de precisión de nuestro sistema es igual a $(VP + VN) / N$ y la tasa de error de $(FP + FN) / N$.

Los resultados obtenidos en esta métrica han sido $VP = 19,597$, $VN = 161,614$, $FP = 14,935$ y $FN = 3,854$. Por lo que la precisión ha sido de 90.61% y la Tasa de Error = 9.39%.

En la tabla 1 mostramos los resultados obtenidos con los distintos métodos supervisados. Estos modelos los hemos creado con el paquete Caret de R Studio y con el método “repeatedcv”, con 10 particiones y 3 repeticiones. Estos modelos se han creado con 800,000 muestras y la precisión del modelo se ha medido con 200,000 muestras del mismo modo que las redes ROA.

Tabla 1. Nuestra propuesta del modelo de predicción CTR basado en las redes de hidrocarburos artificiales.

	R. Orgánicas	Árboles C5.0	CART	CART Costo-sensible	Splines de reg mult. de adap	Cent dism más cercano	Reduc. análisis discr.	Análisis discrim lineal	Penal. en análisis discrim.	OneR
Precisión	0,9061	0,8407	0,8394	0,8392	0,8384	0,8379	0,8374	0,8373	0,8367	0,832
Tiempo construir modelo	0:00:12	0:00:26	0:00:10	0:00:26	0:00:28	0:00:03	0:00:03	0:00:02	0:00:05	0:00:07

En la Figura 2 mostramos la precisión de cada uno los modelos en comparación con el promedio de todos los métodos.

5. Conclusión

Consideramos que las ROA son un modelo de datos muy efectivo y que se pueden adaptar a muchos tipos de problema. Además tienen la ventaja de que permiten crear modelos en un tiempo reducido.

Las ROA permiten una predicción de CTR de aproximadamente el 90% de precisión lo cual es un buen resultado. Una mejora podría ser aplicar una heurística que mejorara los resultados de la estimación por mínimos cuadrados. También se podrían utilizar técnicas híbridas de aprendizaje (p.e., redes bayesianas para datos incompletos, árboles AVL para mejorar la búsqueda de información, etc.), ya que una sola técnica limita el potencial de aprendizaje en problemas reales.

Una desventaja de las ROA es que no soportan ser entrenadas por un gran número de muestras. Una solución para esto sería entrenar un conjunto redes orgánicas. Por ejemplo crear 1,000 redes orgánicas artificiales y predecir el resultado como el promedio de todas las redes.

Por otra parte, consideramos que este tipo de redes pueden ser un buen aliado en la lucha contra el fraude que se produce en la publicidad en Internet. Existen muchas

amenazas para las redes como son los *botnets*⁷ [12], las granjas de clics⁸ o los propios editores⁹. Todas las redes de publicidad están expuestas a este tipo de amenazas y muchas de ellas son muy difíciles de detectar. Las ROA son muy útiles para la detección basada en anomalías. Es decir, en primer lugar se podrían entrenar las redes con parámetros que representen el comportamiento habitual de los usuarios. Una vez entrenadas, cuando las redes perciban un comportamiento distinto al acostumbrado podrán mandar un mensaje de alarma.

La principal ventaja que tienen las ROA en este campo es que son capaces de señalar el parámetro por el cual han considerado la visita no es válida. Es decir, que además de descartar el clic nos proporciona cierta información sobre esta decisión. Junto con las ROA, sería conveniente utilizar algunos filtros que comprueben si existen demasiadas IPs de la misma zona, usuarios reincidentes o un excesivo número de clics en un determinado anuncio.

6. Trabajos futuros

Las ROA son modelos predictivos muy eficaces y que tienen utilidad para un gran número de problemas. Un proyecto interesante con este tipo de redes sería crear un paquete para R Studio o para Matlab. Esto facilitaría que muchas personas las usaran y pudiesen comprobar su precisión frente a otros modelos de una forma sencilla.

Otra mejora consistiría en crear un conjunto de predictores para aumentar el porcentaje de aciertos o de predicciones tanto para clasificación como para regresión. En el caso de clasificación la clase vendrá determinada por la clase que predigan la mayoría de los modelos. En el caso de los modelos de regresión, el valor que se prediga será el promedio de los valores que predigan todos los modelos.

Las redes orgánicas artificiales pueden ser muy útiles para la detección de fraude en la publicidad en Internet. Estas redes podrían entrenarse con visitas normales realizadas por los usuarios y en el caso de que detectaran un comportamiento anómalo avisaran a la red publicitaria. También se puede hacer a la inversa, es decir, entrenar las redes con las visitas de los usuarios fraudulentos y en caso de que sea parecido notificar a la red de publicidad.

Otra línea de investigación futura podría ser hacer una estimación del tiempo necesario para construir el modelo con ROA. De este modo, si el tiempo es muy alto el usuario podrá abortar la operación o planificarse teniendo en cuenta el tiempo de cálculo.

Para ello, deberíamos crear un modelo con varias muestras que tuviera como entradas: el número de muestras, el número de procesadores, la memoria RAM, la velocidad de los procesadores, el sistema operativo y el número de entradas. Este algoritmo se podría ir actualizando según se vaya ejecutando el programa de la misma

⁷ Los botnets son pequeños programas que se insertan en las computadoras de los usuarios y que pueden navegar y hacer clics sin que el usuario lo sepa.

⁸ Las granjas de enlaces son equipos de personas que pueden estar en países como la India y que se dedican a hacer clic para arruinar campañas de publicidad.

⁹ Los editores pueden hacer trampas para aumentar sus propios ingresos y muchas veces piden ayuda a otras personas para hacerlo.

forma que lo hacen las descargas. Pues no tenemos garantizado que el 100% del procesador vaya a ser para este programa.

Referencias

1. Coopers, P. W. H., IAB internet advertising revenue report. URL: http://www.iab.net/insights_research/industry_data_and_landscape/adrevenue-report (2014)
2. Moe, W. W.: Targeting display advertising. *Advanced database marketing: Innovative methodologies & applications for managing customer relationships*. Londres: Gower Publishing (2013)
3. Stone-Gross, B., Stevens, R., Zarras, A., Kemmerer, R., Kruegel, C., Vigna, G.: Understanding fraudulent activities in online ad exchanges. In: *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*, pp. 279–294, ACM (2011)
4. McMahan, H. B., Holt, G., Sculley, D., Young, M., Ebner, D., Grady, J. Kubica, J.: Ad click prediction: a view from the trenches. In: *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 1222–1230, ACM (2013)
5. Tuzhilin, A.: The lane's gifts v. google report. Official Google Blog: Findings on invalid clicks. pp. 1–47 (2006)
6. Ponce, H., Ponce, P., Molina, A.: *Artificial Organic Networks: Artificial Intelligence Based on Carbon Networks*. *Studies in Computational Intelligence*, Vol. 521, Springer (2014)
7. Ponce, H., Ponce, P., Molina, A.: A New Training Algorithm for Artificial Hydrocarbon Networks Using an Energy Model of Covalent Bonds. In: *7th IFAC Conference on Manufacturing Modelling, Management, and Control*, Vol. 7(1), pp. 602–608 (2013)
8. Ponce, H., Ponce, P.: *Artificial Organic Networks*. In: *IEEE Conference on Electronics, Robotics, and Automotive Mechanics CERMA*, pp. 29–34 (2011)
9. Kuhn, M.: Building predictive models in R using the caret package. *Journal of Statistical Software*, 28(5), 1–26 (2008)
10. Granitto, P. M., Furlanello, C., Biasioli, F., Gasperi, F.: Recursive feature elimination with random forest for PTR-MS analysis of agroindustrial products. *Chemometrics and Intelligent Laboratory Systems*, 83(2), 83–90 (2006)
11. Kuhn, W., Wing, J., Weston, S., Williams, A., Keefer, C., et al.: Caret: classification and regression training. R package, v515 (2012)
12. Miller, B., Pearce, P., Grier, C., Kreibich, C., Paxson, V.: What's clicking what? Techniques and innovations of today's clickbots. In: *Detection of Intrusions and Malware, and Vulnerability Assessment*, pp. 164–183, Springer (2011)

Análisis del desempeño del algoritmo genético en la clasificación automática de documentos

Juan Manuel Zárate Sánchez, Hilarión Muñoz Contreras,
María Antonieta Abud Figueroa

División de Estudios de Posgrado e Investigación, Instituto Tecnológico de Orizaba,
México

manuel_zasa86@hotmail.com, hmunozc189@msn.com
mabud@ito-depi.edu.mx

Resumen. Existen diversos algoritmos de clasificación para el proceso de clasificación automática de documentos, estos algoritmos buscan soluciones eficientes y rápidas, es por lo cual, el Algoritmo Genético es ideal para aplicar a este contexto, ya que es capaz de encontrar soluciones eficientes en unos cuantos segundos, por su capacidad de examinar el espacio de búsqueda en forma amplia y eficiente. Este Artículo se centra principalmente en analizar el desempeño del Algoritmo Genético para la clasificación automática de documentos. En este artículo se analizó el desempeño que tiene el algoritmo OlexGA, que posee como base un Algoritmo Genético, mediante diversas pruebas de minería de datos.

Palabras clave: algoritmo genético, clasificación texto, algoritmo evolutivo, OlexGA, minería de datos, WEKA.

1. Introducción

La clasificación automática de documentos se define como el proceso de separar los documentos en grupos o clases. El criterio de agrupamiento es acuerdo en las similitudes que existen entre ellos [1]. La clasificación automática de documentos se relaciona con dos disciplinas Informáticas :1) la Minería de Datos y 2) la Recuperación de Información, la Primera disciplina utilizan las técnicas que se ocupan para la clasificación de objetos, por último, la segunda disciplina utilizan los conceptos y métodos para el pre-procesamiento de documentos, es decir, esto consiste en la transformación de la información no estructurada de los documentos a información estructurada manejable para los algoritmos de clasificación [2].

Este artículo está estructurado de la siguiente manera. En la sección 2 se analizó el estado del arte de la clasificación automática de documentos aplicando un enfoque evolutivo. En la sección 3 se proporciona una breve descripción del proceso de clasificación automática de documentos. La sección 4 aborda como se realiza las pruebas, en la sección 5 se muestran los resultados de las pruebas, en la sección 6 se presentan la discusión de los resultados y por último, la sección 7 se tiene las conclusiones.

2. Estado del arte

En el contexto de la Minería de Datos, el uso de Algoritmo Genético para clasificar documentos ofrece múltiples ventajas como ahorro de tiempo, dinero y esfuerzo. Por tal razón, diversos trabajos en la literatura han abordado este enfoque.

Yolis et al. [3] describen las restricciones que presentan los algoritmos de clasificación para categorizar los documentos, argumentando que los tiempos de cómputo son inaceptables. Según la perspectiva del autor, una solución para resolver esta problemática es la aplicación del Algoritmo Genético. Este trabajo propone un algoritmo genético que presenta un operador cruzado y cuatro operadores de mutación, diseñados especialmente para solucionar la problemática de la clasificación automática de documentos, el diseño del algoritmo de clasificación se adapta a los conceptos que aplican los algoritmos genéticos. Se realizaron experimentos cuyos resultados confirman que el algoritmo propuesto es una buena solución para la clasificación de documentos. Teniendo como conclusión que los algoritmos genéticos son una herramienta poderosa para la resolución de problema donde las soluciones son amplias y optimización son complejas, sin embargo, el autor afirma que los algoritmos genéticos no son un método de solución universal de problemas, sino un paradigma que debe adaptarse correctamente al problema a resolver.

K. Premalatha et al. [4] describen un proceso de clasificación de documentos basado en el Algoritmo Genético con un operador de mutación simultánea y un número de tasa de mutación. En la mutación simultánea, el algoritmo genético, utiliza varios operadores de mutación en la producción de la próxima generación. El operador de mutación es representativo para el éxito de los algoritmos genéticos, ya que amplían las direcciones de búsqueda y evita la tendencia a una optimización local. El algoritmo propuesto aumentó notablemente el éxito para la clasificación de documentos, para comprobar esto se comparó con el algoritmo genético simple con el algoritmo de K-mean.

Meena et al. [5] exponen que las técnicas de agrupación tienen mejor optimización aplicando el enfoque del Algoritmo Genético. El algoritmo k-mean es un método popular para la categorización de documentos de texto, pero sus resultados se basan en la elección de los centros de conglomerados, esto fácilmente se traduce en optimización local. En este trabajo se propuso en utilizar un método dinámico para la agrupación basado en el Algoritmo Genético, el objetivo es encontrar fácilmente mejores centros de conglomerados y reduciendo las iteraciones mediante con otra técnica de optimización DDE (*Discrete Differential Evolution*). Se obtuvo como conclusión que hay una mejora en la categorización de documentos utilizando la combinación del algoritmo genético y DDE.

A. Casillas et al. [6] presentan un algoritmo que realiza la categorización de una colección de documentos, sin conocer previamente el número de agrupación. En este trabajo se diseñó e implementó un algoritmo genético que encuentra un valor próximo al óptimo número de agrupación k , con un costo computacional menor que con la regla de Calinski y Harabasz, este algoritmo obtiene un valor aproximado del óptimo número de agrupación k y resuelve la forma de agruparlos dentro de las agrupaciones de k . El autor evaluó este algoritmo con conjuntos de documentos, que son el resultado de salida de un motor de búsqueda ante consultas de un usuario, por lo cual se afirma que el algoritmo es capaz de realizar una agrupación en línea.

A. Villagra et al. [7] detallan en forma breve una de las líneas de investigación que se realizó en el Laboratorio de Tecnologías Emergentes (LabTEm) sobre Algoritmos Evolutivos y su aplicación como técnica alternativa y complementaria en tareas de Minería de Datos, concretamente en la clasificación de objetos. En la actualidad, se almacenan enormes cantidades de datos y no es fácil representar e interpretar los datos que están almacenados, por consiguiente es difícil obtener información relevante para la toma de decisiones basadas en dichos datos. La Minería de Datos implica "escavar" en esa inmensidad de datos, en búsqueda de patrones, asociaciones o predicciones que permitan transformar esa maraña de datos en información útil. Esta línea de investigación tuvo como resultado técnicas avanzadas de minería de datos basadas en el enfoque evolutivo.

Veronica et al. [8] presentan un Algoritmo Genético con el nombre Olex-GA, para clasificación de texto basados en reglas de inducción de la siguiente forma: "clasificación de documentos d bajo la categoría c si $t_1 \in d$ o ... o $t_n \in d$ y no contenga ($t_{n+1} \in d$ o ... o $t_{n+m} \in d$)", donde cada t_i es un término. Olex-GA se basa en una eficiente representación binaria de varias reglas por individuos y utiliza como función de aptitud F-medida. El algoritmo de clasificación propuesto fue probado sobre la prueba estándar de los conjuntos Reuters 21578 y Ohsumed, y se comparó con varios algoritmos de clasificación.

Como se observa en la revisión del estado del arte, la utilización de Algoritmo Genético en la minería de datos es una buena solución para la clasificación automática de documentos.

3. Proceso de clasificación automática de documentos

Para comprender el proceso de clasificación automática de documentos se debe analizar el proceso normal que hace un clasificador humano; este proceso se define de la siguiente forma:

"... el acto de organizar el universo del conocimiento en algún orden sistemático. Se considera la actividad más importante de la mente humana. El hecho de clasificar reside en el dicotómico proceso de diferenciar los objetos que tienen ciertas características de otros objetos que no tienen y agruparlos en una clase los objetos que tienen características comunes" [9].

Esto se traduce en términos de gestión documental, que la clasificación de documentos manual consiste en tres simples pasos:

- Leer el documento para tener una idea precisa de su contenido.
- Resumir el contenido en un tema principal
- Establecer qué categoría se acerca más al tema principal y representar el documento mediante una notación propia de la clasificación, de tal forma que el almacenamiento esté en orden y que la recuperación del documento sea posible.

El clasificador automático no tiene la capacidad de enfrentar en una forma directa algunas tareas de la clasificación manual, por lo cual estas tareas son realizadas con diferentes métodos que lleven a un resultado similar. Por ejemplo, la tarea de leer el documento e identificar su contenido, el sistema de clasificador automático suele llevar a cabo este proceso mediante la utilización de métodos de indización automática. Es

evidente que el proceso de indización automática no tiene nada que ver con el proceso que realiza un clasificador humano, ya que este tiene una lectura secuencial, además, el sistema de clasificación automática no entiende o comprenden los contenidos de los documentos, pero lo representa en una forma que simule el entendimiento.

En la tarea de resumir el contenido en un tema principal, es un proceso fácil para un clasificador humano, pero para el sistema de clasificación automática es lógico pensar que no pueda asignar un tema principal a un contenido de un documento, como lo haría un clasificador humano, ya que el sistema de clasificación automático no comprende el contenido y desconoce el tema principal o su significado.

Por último, el proceso de establecer la clasificación de un documento de acuerdo con su tema principal es una tarea relativamente sencilla para un clasificador humano, ya que este comprobaría la pertenencia del tema principal del documento a cada una de las categorías principales de la clasificación hasta encontrar la correcta y después descender por las subcategorías hasta localizar la que más se adapte al tema principal del documento. Un clasificador automático no podrá llevar a cabo este proceso de forma directa, ya que no conoce el significado del tema principal del documento.

Por lo anterior descrito, es evidente que el trabajo de un clasificador humano no se logra llevar a cabo directamente por un clasificador automático, además, la parte inteligente del sistema de clasificación automático está enfocado en la aplicación de un algoritmo que permite el aprendizaje.

Los algoritmos de clasificación asignan a los documentos una o varias etiquetas, categorías o clases preestablecidas, sin embargo, no tiene como objetivo emular el comportamiento de un clasificador humano, sino aprovecharse de él, para lograr las mismas conclusiones con diferentes métodos. Para lograr aprovecharse del clasificador humano es necesario utilizar un conjunto de documentos de entrenamiento. Este conjunto de documento de entrenamiento ha sido preclasificado por clasificadores humanos expertos en la materia.

El proceso de clasificación automática de documento suele contener tres etapas generales [10] :

1. Pre-procesamiento,
2. Construcción de Clasificador,
3. Clasificación de nuevos documentos.

A continuación se describirán las etapas de la categorización automática de documentos.

3.1.Preprocesamiento

En esta etapa de proceso de clasificación automática de documentos, los textos de los documentos se trasfiere a un formato compacto que se utiliza en las siguientes etapas. En este proceso incluye los siguientes métodos:

Stemming: Es un método reduce una palabra a su raíz, por ejemplo, tenemos una consulta sobre “bibliotecas” que también saldrán documentos que tenga la palabra “bibliotecario”, porque la raíz de las dos palabra es “bibliotec”.

Stopwords: es una técnica que elimina palabras de función, por ejemplo: preposiciones, artículos, conjunciones y otras palabras dependiente del dominio. Esta eliminación es acuerdo a un diccionario preestablecido y se aplica después del método de *Stemming*

Elección de atributos: cada palabra del texto será un atributo, esto solo tendrá dos valores 1 y 0, las palabras que sean irrelevante serán eliminadas, por ejemplo, las palabras que se repitan en todos los documentos.

Con estos métodos se construye la representación compacta de los documentos para los algoritmos de clasificación.

3.2.Construcción de clasificador

La gran parte de los algoritmos de clasificación llevan a cabo un tipo de entrenamiento o aprendizaje antes de realizar la tarea de clasificar nuevos documentos. Para realizar este tipo de aprendizaje se debe contar con un conjunto de documentos de entrenamiento ya clasificados por expertos. Los conjunto de documentos de entrenamiento son los documentos convertidos en conjuntos de atributos y valores por la etapa anterior, por lo cual, se utilizan para construir un clasificador que asignará categorías a nuevos documentos.

3.3.Clasificación

Este el último proceso de la clasificación automática de documentos, es la más rápida que los anteriores procesos, ya que todo el trabajo del proceso de clasificación automática de documentos recae en la etapas de pre-procesamiento y construcción del clasificador. Para clasificar un conjunto de documentos nuevos, previamente pre-procesado, se utiliza el clasificador construido en la anterior etapa.

4. Análisis del desempeño del clasificador basado en algoritmo genético

Una vez que se comprendió el proceso de Clasificación Automática de Documentos se procedió a la evaluación del desempeño del Algoritmo Genético para la clasificación de documentos. Cabe mencionar que este trabajo no tiene como objetivo desarrollar un sistema de clasificación de documentos, sino realizar un análisis del desempeño del Algoritmo Genético para la tarea de clasificación de texto y comparando con otros algoritmos de clasificación, por lo cual, se utilizó API (Application Programming Interface) de WEKA.

Los criterios para medir el desempeño del algoritmo genético giran en dos criterios: 1) El criterio de eficiencia hace referencia a las medidas que están relacionados con los tiempos computacionales requeridos para llevar a cabo los procesos construcción del clasificador y la evaluación del clasificador. 2) El criterio de efectividad se refiere a nuestra capacidad para medir el comportamiento que tiene el clasificador desde un punto de vista de la calidad de resultados obtenidos por él mismo. Para medir el desempeño de los algoritmos de clasificación se emplearon 3 tipos de prueba:

Percentage Split: En este caso, se dividirá el conjunto de entrenamiento en dos partes: los primeros 66% de los datos para construir el clasificador y el 33% finales, para hacer la evolución.

Cross-validation: Este tipo de prueba se realiza una validación cruzada estratificada del número de particiones dado (*Folds*). La validación cruzada consiste en: dado un número n se divide los datos en n partes y, por cada parte, se construye el clasificador con las $n-1$ partes restantes y se prueba con esa partición.

Supplied test set: Este tipo de evaluación se construye el clasificador con un conjunto de documentos de entrenamiento y el clasificador se evalúa con otro conjunto de documentos de prueba para calcular sus aciertos.

4.1. Conjunto de documentos

En este trabajo se usó el conjunto de documentos denominado *Reuters 21758*, es una colección de documentos de noticias reales que aparecieron en cables de la agencia Reuters durante 1987. Los documentos fueron recopilados y categorizados manualmente por personal de la agencia y de la compañía Carnegie Group [11]. La colección *Reuters 21578* es un estándar para la prueba de algoritmos de clasificación, por lo que los resultados obtenidos tendrán mucha más validez que si los experimentos se realizaran sobre un conjunto de datos recopilados sin seguir una metodología estándar. Además se utilizó en diversos trabajos [4], [12], [13], [14], [15], [16] y [17]. Antes de usar la colección de Reuters 21578 se tiene que realizar un pre-procesamiento para obtener una representación compacta de los documentos, en este trabajo se utilizó la representación compacta de un investigador del El Instituto de Teoría de la Información y Automatización de la Republica Checa, que lo puso a disposición en [18].

Esta representación está dividida en dos conjuntos: el primer conjunto de documentos de entrenamiento, que contienen 7769 documentos y el segundo es de conjunto de documentos de prueba (*test*), que contienen 3018 documentos, además, sólo se seleccionó diez clases que contienen la mayor cantidad de documentos para la prueba, estas clases se observan en la tabla 1.

Tabla 1. Las diez clases o clasificación con mayor cantidad de documentos de prueba

Clase	Número de documentos de esta clase en el conjunto de pruebas
EARN	1087
ACQ	719
CRUDE	189
MONEY-FX	179
GRAIN	149
INTEREST	131
TRADE	117
SHIP	89
WHEAT	71
CORN	56

Todos los conjuntos documentos entrenamiento y de prueba de la colección de Reuters se convirtió al formato ARFF, que es un formato de WEKA, para construir los diferentes clasificadores.

4.2. Algoritmos de clasificación

Existen diversos algoritmos de clasificación empleados para la clasificación de documentos, sin embargo, no son específicamente para este fin sino que se han propuesto para clasificar de todo tipos de objetos, es decir, algunos de esto se adaptan con más o menos para la clasificación de documentos. A continuación se muestran los algoritmos de clasificación empleados en los experimentos, cabe aclarar que estos algoritmos son los que han sido utilizados para la clasificación de documentos y están implementados en WEKA: OlexGA, Ridor, JRip, J48, ADTree, IBK, NaiveBayesMultinomial y LibSVM.

El Algoritmo OlexGA es un algoritmo de clasificación basado en el Algoritmo Genético, la descripción de este algoritmo se encuentra en [8], el cual se va analizar su desempeño por ser un algoritmo con enfoque evolutivo.

5. Resultados

En esta sección se presentan los resultados obtenidos de la realización de las pruebas descritas anteriormente, cada prueba se repitió 10 veces para sacar un promedio de cada algoritmo de clasificación.

En la prueba de *Percentage Split* se obtuvo los datos que se muestran en las tablas 2 y 3, con estos resultados se llega a la conclusión que el algoritmo OlexGA tiene en promedio de precisión de 96.14, sin embargo, algunos algoritmos tienen un promedio de precisión mayor que OlexGA, por ejemplo, JRIP Y NaiveBayesMultinomial, pero su promedio de tiempo es mayor que el OlexGA, por lo cual, este algoritmo de clasificación con enfoque evolutivo tiene un buen desempeño al dar resultados aceptables en pocos segundos (ver Tabla 3).

Tabla 2. Precisión obtenida de la prueba de *Percentage Split*.

Clases	OlexGA	Ridor	JRip	J48	ADTree	IBk	NaiveBayesM.	LibSVM
ACQ	86.51	92.33	93.64	93.76	90.07	93.66	96.54	94.71
CORN	99.53	99.38	99.56	99.41	99.45	98.33	98.29	95.28
CRUDE	97.38	97.27	97.86	97.76	96.98	97.36	97.5	96.49
EARN	95.53	95.69	95.85	95.51	94.11	95.59	95.66	96.14
GRAIN	98.89	98.46	98.87	98.54	98.55	96.74	97.01	96.56
INTEREST	96.59	96.8	97.28	96.88	96.35	97.32	97.37	96.08
MONEY	96.84	96.21	97.13	96.57	95.63	97.06	96.63	95.8
SHIP	98.81	98.04	98.56	98.42	98.34	98.36	98.72	96.83
TRADE	92	96.61	97.12	96.95	96.56	96.61	96.38	96.15
WHEAT	99.33	99.38	99.47	99.33	99.44	97.79	97.97	94.91
PROMEDIO	96.14	97.01	97.53	97.31	96.54	96.88	97.20	95.89

Tabla 3. Tiempo obtenido para la construcción del clasificador y su evaluación en la prueba de *Percentage Split*.

Clases	OlexGA	Ridor	JRip	J48	ADTree	IBk	NaiveBayesM	LibSVM
ACQ	4.928	30.89	64.11	29.1	47.01	40.63	4.126	6.912
CORN	5.457	6.969	10.6	4.486	41.82	70.44	7.117	2.696
CRUDE	4.715	7.149	24.96	11.67	49.43	60.59	6.121	3.015
EARN	4.535	28.02	30.9	16.7	45.97	50.07	5.056	6.82
GRAIN	4.617	7.839	19.51	12.71	51.46	52.82	5.341	3.196
INTEREST	5.748	8.53	31.72	16.21	48.42	47.33	4.795	3.467
MONEY	4.714	9.27	28.8	20.78	46	55.5	5.619	4.494

Clases	OlexGA	Ridor	JRip	J48	ADTree	IBk	NaiveBayesM	LibSVM
SHIP	5.298	10.15	18.94	15.4	51.55	45.33	4.59	1.743
TRADE	5.149	8.27	21.95	9.387	45.86	56.63	5.723	3.477
WHEAT	6.481	5.475	8.764	5.754	37.49	58.4	5.918	2.808
PROMEDIO	5.1642	12.25	26.02	14.21	46.50	53.77	5.44	3.86

En la prueba de *Cross-validation* se obtuvo los siguientes resultados que se muestran en las tablas 4 y 5, con estos datos se llega a la conclusión que el algoritmo OlexGA tiene un promedio de precisión aceptable en un menor tiempo que los otros algoritmos de clasificación analizados.

Tabla 4. Precisión obtenida de la prueba de *Cross-validation*.

Clases	OlexGA	Ridor	JRip	J48	ADTree	IBk	NaiveBayesM.	LibSVM
ACQ	86.7	92.45	93.65	93.85	90.27	93.63	96.46	94.75
CORN	99.6	99.44	99.55	99.48	99.51	98.36	98.25	95.39
CRUDE	97.56	97.22	97.79	97.62	96.95	97.23	97.52	96.53
EARN	95.5	95.28	95.83	95.46	94.07	95.76	95.85	96.22
GRAIN	98.89	98.51	98.88	98.52	98.64	96.69	96.95	96.51
INTEREST	96.59	96.81	97.21	96.8	96.41	97.35	97.38	95.96
MONEY	96.78	96.29	97.04	96.47	95.6	96.97	96.53	95.88
SHIP	98.77	98.09	98.58	98.45	98.33	98.34	98.75	97.58
TRADE	92.31	96.76	97.24	97.07	96.6	96.63	96.34	96.34
WHEAT	99.41	99.4	99.45	99.38	99.41	97.91	97.99	95.43
PROMEDIO	96.211	97.02	97.52	97.31	96.579	96.88	97.20	96.05

Tabla 5. Tiempo obtenido para la construcción del clasificador y su evaluación en la prueba de *Cross-validation*

Clases	OlexGA	Ridor	JRip	J48	ADTree	IBk	NaiveBayesM.	LibSVM
ACQ	19.96	202.5	409	239.4	353.3	151.4	15.38	31.55
CORN	23.18	28.14	42.92	18.85	158.6	176.7	17.91	8.089
CRUDE	21.03	55.8	158.1	76.82	252.6	175.7	17.8	11.16
EARN	24.21	95.28	95.83	95.46	94.07	95.76	95.85	96.22
GRAIN	24.01	78.41	178.5	111.9	320.6	159.3	16.14	12.58
INTEREST	22.77	59.95	176.9	102.8	248.2	156.6	15.89	13.5
MONEY	18.73	60.89	179.2	123.6	228.3	179.6	18.19	18.82
SHIP	19.05	71.03	130.1	104.2	273.9	153.1	15.53	6.886
TRADE	25.4	62.26	195.5	84.96	234.2	182.2	18.43	14.09
WHEAT	18.14	24.14	43.7	24.41	127	168.4	17.06	8.334
PROMEDIO	21.648	73.84	160.97	98.24	229.07	159.87	24.81	22.12

En la prueba de *Supplied test set* se obtuvo los siguientes datos que se muestran en las tablas 6 y 7. En la tabla 7 se observa que el algoritmo OlexGA tiene un promedio de tiempo de 5 segundos, es evidente que el algoritmo de clasificación con enfoque evolutivo tiene un mejor tiempo que los demás algoritmos, además, en el promedio de precisión tiene un resultado aceptable.

Tabla 6. Precisión obtenida de la prueba de *Supplied test set*.

Clases	OlexGA	Ridor	JRip	J48	ADTree	IBk	NaiveBayesM.	LibSVM
ACQ	86.4	92.93	94.61	94.7	89.96	93.64	97.21	94.72
CORN	99.49	99.65	99.68	99.71	99.73	98.62	98.36	96.32
CRUDE	97.76	97	97.62	97.42	97.46	95.67	97.24	95.95
EARN	97.33	96.57	96.99	96.61	96.34	97.38	97.68	98.05
GRAIN	99.25	98.45	99.11	98.73	98.92	97	97.18	97.54
INTEREST	96.38	96.68	97.25	96.15	96.66	96.68	97.25	96.15
MONEY	96.31	95.84	96.46	95.93	95.11	95.39	96.6	95.38

Clases	OlexGA	Ridor	JRip	J48	ADTree	IBk	NaiveBayesM.	LibSVM
SHIP	98.61	98.14	98.81	98.58	98.89	96.6	98.71	97.41
TRADE	93.65	97.44	97.73	97.5	97.57	96.34	96.91	96.95
WHEAT	99.53	99.53	99.57	99.39	99.52	98.29	98.21	96.55
PROMEDIO	96.47	97.22	97.78	97.47	97.01	96.56	97.53	96.50

Tabla 7. Tiempo obtenido para la construcción del clasificador y su evaluación en la prueba de *Supplied test set*.

Clase	OlexGA	Ridor	JRip	J48	ADTree	IBk	NaiveBayesM.	LibSVM
ACQ	6.042	68.55	125	57.55	85.39	62.87	6.37	12.08
CORN	4.603	9.733	15.18	6.009	62.04	77.52	7.828	3.617
CRUDE	4.502	19.92	61.93	30.13	90	81.28	8.204	5.008
EARN	4.753	47.7	60.45	32.81	81.07	77.07	7.799	16.13
GRAIN	5.729	30.81	70.07	45.09	134.2	83.55	8.443	7.764
INTEREST	5.903	25.75	84.1	46.82	84.63	64.73	6.546	6.001
MONEY	4.684	26.57	77.18	48.86	82.64	74.32	7.507	7.806
SHIP	4.557	23.73	55.56	40.76	95.39	59.73	6.047	2.943
TRADE	4.876	19.25	49.8	23.88	79.34	73.35	7.408	6.339
WHEAT	4.401	9.17	16.98	10.05	39.34	74.64	7.538	3.807
PROMEDIO	5	28.11	61.62	34.19	83.404	72.90	7.36	7.14

6. Discusión de los resultados

Este trabajo tuvo la finalidad de analizar el desempeño que tiene el Algoritmo Genético para el problema general de la clasificación automática de documentos aplicando las técnicas de minería de datos que ofrece la herramienta de WEKA. Los resultados obtenidos de las tres pruebas, se puede deducir que se obtienen resultados aceptables, como se muestra en la figura 1.

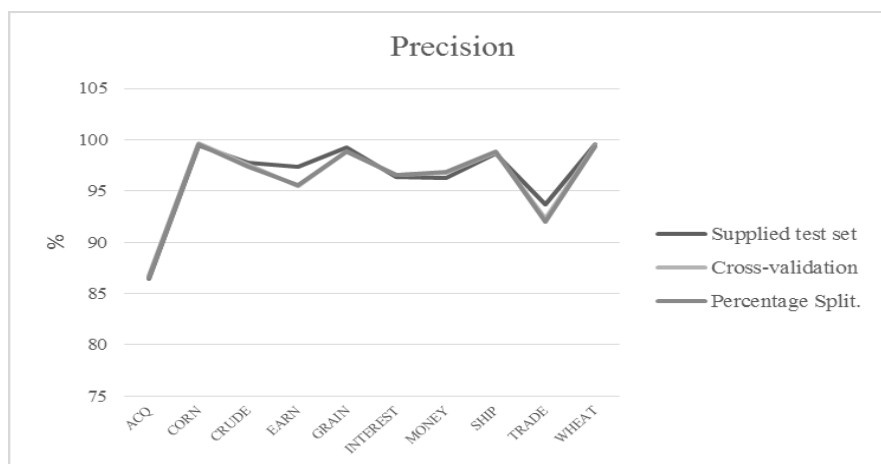


Fig. 1. Grafica de los resultados de presión en las tres pruebas.

Además se consigue estos resultados en un tiempo menor que la mayoría de los algoritmos de clasificación analizados, como se muestra en la figura 2, por lo cual, se llega a la conclusión que la aplicación de los Algoritmos Genético en la clasificación de automática de documentos se consiguen soluciones aceptables en un menor tiempo.

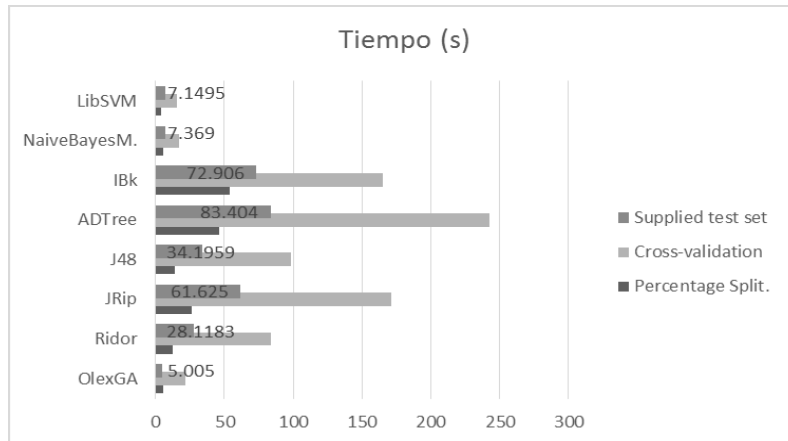


Fig. 1. Tiempo promedio de los algoritmos de clasificación en las tres pruebas.

7. Conclusión

Los algoritmos de clasificación de documentos están en creciente atención debido al aumento de la cantidad de información electrónica, y a la necesidad de accederla en el menor tiempo posible y con mayor eficacia. Si bien existen algoritmos que dan resultados aceptables en la clasificación de documentos, los tiempos de cómputo que requieren son inaceptables para las aplicaciones prácticas.

En las pruebas realizadas para este trabajo se confirma que los algoritmos genéticos son una poderosa herramienta para la solucionar el problema general de la clasificación automática de documentos, dando como resultado soluciones aceptables en un mejor tiempo.

Referencias

- [1] Kaufman, L., Rousseeuw, J.: Finding groups in data: an introduction to cluster analysis. In: Applied Probability and Statistics, New York, Wiley Series in Probability and Mathematical Statistics (1990)
- [2] Yolis, E.: Algoritmos genéticos aplicados a la categorización automática de documentos. Facultad de ingeniería universidad de Buenos Aires, Buenos Aires, (2003)
- [3] Yolis, E., Britos, P., Sicre, J., Servetto, A., García-Martínez, R., Perichinsky, G.: Algoritmos genéticos aplicados a la categorización automática de documentos. In: IX Congreso Argentino de Ciencias de la Computación, pp. 1468–1479 (2003)
- [4] Premalatha, A.M., Natarajan, K.: Genetic Algorithm for Document Clustering based on Simultaneous and Ranked Mutation. Journal of Modern Applied Science, vol. 3, no. 2, pp. 35–42 (2009)
- [5] Meena, K., Singh, P.: Text Documents Clustering using Genetic Algorithm and Discrete Differential Evolution. International Journal of Computer Applications, vol. 43, no. 1 (2012)

- [6] Casillas, A., González de Lena M., Mart, R.: Document Clustering into an unknown number of clusters using a Genetic Algorithm. In: *Speech and Dialogue: 6th International Conference*, vol. 6, pp. 43–49 (2003)
- [7] Villagra, A., Pandolfi, D., Lasso, M., de San Pedro, M.: Algoritmos evolutivos y su aplicabilidad en la tarea de clasificación. In: *VIII Workshop de Investigadores en Ciencias de la Computación* (2006)
- [8] Pietramala, A., Policicchio, V. L., Rullo, P., & Sidhu, I.: Genetic algorithm for text classification rule induction. In: *Machine Learning and Knowledge Discovery in Databases*, pp. 188–203 (2008)
- [9] Chan, M.L.: *Cataloging and classification: an introduction*. New York: McGraw-Hill, p. 209 (1981)
- [10] Abelleira, M., Pérez, A.: *Minería de texto para la categorización automática de documentos*. PhD in Computer Science por Carnegie Mellon University, Madrid, España (2010)
- [11] U. of California, UCI Knowledge Discovery in Databases Archive, University of California, Irvine, 9 Septiembre 2005. [En línea]. Available: <http://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html>. [Último acceso: 2014 Septiembre 01].
- [12] Jiménez, R. S.: La documentación en el proceso de evaluación de Sistemas de Clasificación Automática. *Documentación de las Ciencias de la Información*, vol. 30, pp. 25–44 (2007)
- [13] Goldenberg, D.: *Categorización Automática de Documentos con Mapas Auto Organizados de Kohonen*. Tesis de Magister en Ingeniería del Software, Universidad Politécnica de Madrid, España, (2007)
- [14] Toutanova, K., Chen, F. R., Popat, K., Hofmann, T.: Text classification in a hierarchical mixture model for small training sets. In: *Proceedings of the tenth international conference on Information and knowledge management*, ACM, pp. 105–113 (2001)
- [15] Tikk, D., Dong-Yang J., Lee-Bang, S.: Hierarchical text categorization using fuzzy relational thesaurus. *Kybernetika*, vol. 30, pp. 583–600 (2003)
- [16] Sciarrone, F.: An extension of the q diversity metric for information processing in multiple classifier systems: a field evaluation. *International Journal of Wavelets, Multiresolution and Information Processing*, vol. 11, no. 6 (2013)
- [17] Peleja, F., Pereira-Lopes, G., Silva, J.: Text Categorization: A comparison of classifiers, feature selection metrics and document representation. In: *Proceedings of the 15th Portuguese Conference in Artificial Intelligence*, pp. 660–674 (2011)
- [18] Karčiauskas, G.: Reuter's data preprocessed by Gytis Karčiauskas, The Institute of Information Theory and Automation, 2014. [En línea]. Available: <http://staff.utia.cas.cz/vomlel/reuters-data.html>. [Último acceso: 02 Octubre 2014].

Sistema multiagente integrador de bibliotecas digitales

Christian Sánchez-Sánchez, Héctor Jiménez-Salazar,
Carlos Rodríguez-Lucatero, Esaú Villatoro-Tello, Gabriela Ramírez-de-la-Rosa

Departamento de Tecnologías de la Información,
División de Ciencias de la Comunicación y Diseño,
Universidad Autónoma Metropolitana Unidad Cuajimalpa, México D.F.

{csanchez, hjimenez, crodriguez, evillatoro, gramirez}@correo.cua.uam.mx

Resumen. Debido a la existencia de múltiples Bibliotecas Digitales en Internet, cuando un usuario pretende obtener una buena parte de la información disponible sobre un tema, éste tiene que invertir mucho tiempo para buscar las fuentes dentro de ellas y después integrar los resultados que considera relevantes. En este artículo se presenta la aplicación de un sistema multiagente que realiza búsquedas de documentos en diferentes bibliotecas digitales dada una determinada consulta y criterio, para que posteriormente integre los resultados y los presente al usuario. El sistema integra las ventajas de los sistemas Multiagente, Arquitecturas Orientadas a Servicios y algunos de los principios de la Integración de Información. En el artículo, se presenta el detalle de la arquitectura propuesta (roles de agentes y protocolos de interacción) y de las fases de Integración de Información, lo que conforma el núcleo del sistema.

Palabras clave: sistema multi-agente, arquitecturas orientadas a servicios, integración de información, bibliotecas digitales.

1. Introducción

Debido a que existen múltiples Bibliotecas Digitales (BD's) en Internet, se ha constatado la complicación de realizar múltiples búsquedas de documentos (libros, revistas, artículos) sobre un determinado tema tratando de aprovechar la mayor parte de la información disponible. El principal problema radica en que el usuario tiene que invertir mucho tiempo en buscar las BD's, dentro de ellas y después integrar los resultados que considera relevantes.

Tratando de resolver este problema la Federación de Bibliotecas Digitales de los Estados Unidos de Norteamérica está desarrollando un proyecto [1] que pretende unir los sistemas web de más de 33 BD's universitarias de ese país junto con los Archivos de la Nación.

Con la finalidad de ofrecer una solución al problema surge una serie de preguntas cuya respuesta puede facilitar la propuesta de un sistema, por ejemplo: 1) *¿Cómo hacer un sistema dinámico que permita incluir y modificar BD's sin tener que reprogramar el sistema?*, 2) *¿Cómo integrar la información de varias*

fuentes?, 3) ¿Cómo re-ordenar (ranking) información ya ordenada de varias fuentes de acuerdo a una consulta y criterio?

Referente a la pregunta 1, hay que considerar que la mayoría de los sistemas de las BD's ya están hechos por lo que la integración de Sistemas es fundamental. Buscando facilitar la integración de los sistemas se pensó en hacer uso de las Arquitecturas Orientadas a Servicios (SOA por sus siglas en inglés) y los Sistemas Multi-agentes (SMA) para lidiar con el dinamismo del sistema.

SOA propone una forma donde el software se presenta como servicios de aplicación. Estos servicios [2] son: débilmente acoplados, altamente interoperables, se pueden re-usar y por ende permiten desarrollar aplicaciones con mayor rapidez.

Por otro lado tenemos al SMA [3] una red débilmente acoplada de agentes de software que interactúan para resolver problemas que van más allá de su conocimiento o capacidades individuales. Es importante señalar que SMA y SOA se pueden complementar. Según Sycara et al. [4] un problema existente se puede resolver mediante un conjunto de servicios (Web o de Agente) los cuales son descubiertos por un agente que integre los resultados de diferentes servicios, lo cual se consideró para proponer una solución a la pregunta 2 y 3.

Con la finalidad de que un agente pueda integrar y re-ordenar información existen algunas metodologías para la integración de información (fusión de datos), un ejemplo de estas son los métodos que según Vogt y Cottrell [5] tienen los siguientes efectos: a) *skimming* que toma los elementos mejor posicionados de cada uno de los enfoques de recuperación, b) *coro* que toma los elementos que en varios enfoques de recuperación se consideran relevantes y c) *caballo negro* se toman algunos elementos cuyas estimaciones de relevancia fueron muy altos o muy bajos, en comparación con otros enfoques de recuperación. Estos efectos son utilizados en la Recuperación de Información (RI), aplicables al tipo de resultados que arrojan las BD's.

En este artículo se presenta un SMA que realiza búsquedas de documentos en diferentes BD's dado una determinada consulta y criterio, para que posteriormente integre los resultados y los presente a un usuario. El sistema integra las ventajas de los SMA, SOA y algunos de los principios de la Integración de Información. En el artículo, se presenta el detalle de la arquitectura propuesta (roles de agentes y protocolos de interacción) y de las fases de Integración de Información, lo que conforma el núcleo del sistema.

El resto del documento se encuentra organizado de la siguiente manera: en la Sección 2 se describe brevemente el trabajo relacionado más relevante a la temática en cuestión. En la Sección 3 se presenta la arquitectura: los participantes (los tipos de agentes) y componentes (servicios web y el registro de servicios). La Sección 4 detalla las fases de la composición de la información integrando los resultados de diferentes BD's. Una breve descripción de la aplicación prototipo se muestra en la Sección 5. Algunas pruebas realizadas al sistema se presentan en la Sección 6. Finalmente, la Sección 7 muestra las conclusiones obtenidas y define las líneas de trabajo futuro.

2. Trabajo relacionado

Existen conjuntos de programas de software diseñados para crear y distribuir colecciones digitales en forma de BD's, tal es el caso de Greenstone [6] que facilita dicha tarea. Sin embargo, el que haya cada vez más BD dificulta la integración de la información que ofrecen.

Según Ibrahim et al. [7] para la integración de BD se plantean varios retos debido a las diferencias prevaletentes en: los tipos de datos que maneja cada fuente de información, la diversidad de los lenguajes de consulta y manipulación de datos, la gran variedad de tipos de información y formatos así como su semántica.

Los autores identifican dos enfoques utilizados para construir sistemas de integración de datos:

Enfoque virtual. El sistema de integración de datos recibe la consulta y la traduce de acuerdo a la fuente de información. Este enfoque es conveniente cuando el número de fuentes de información es muy grande, la información cambia frecuentemente y las necesidades de los clientes es impredecible. Puede ser ineficiente cuando las consultas se generan múltiples veces, las fuentes de información son lentas, costosas o no están disponibles.

Enfoque materializado. La información que se puede extraer de las consultas se obtiene con antelación y es guardada en un *Almacén de Datos* para ser consultada de forma centralizada. Se recomienda este enfoque cuando las porciones de información a consultar sean predecibles y los clientes necesiten un buen desempeño de la consulta.

Seaman et al. [8] establece que para facilitar la integración y contar con contenidos más ágiles, es necesario contar con metadatos que se puedan combinar con otros elementos de las BD. Los autores de este trabajo proponen algunos estándares para usarse como metadatos como Open Archives Initiative (OIA), Metadata Encoding and Transmission Standard (METS), CrossRef, DOI entre otros, que ayuden a ligar la información de la BD. Poniendo énfasis que representa un gran reto pero a la vez una gran oportunidad con muchas ventajas cuando exista una implementación adecuada.

Con lo que respecta a las propuestas del desarrollo de prototipos que integran las BD, Lima et al. [9] proponen un sistema cuya arquitectura recae en un SMA. Los autores apostaron por la interoperabilidad semántica, por medio del uso de relaciones inter-ontológicas y una metabase (base de datos de metadatos) que ayudan a describir información en las BD's. Afirman que cuentan con un novedoso servicio de razonamiento ontológico que representa un modelo conceptual sofisticado de términos y relaciones, que captura la semántica de una manera prometedora para la integración de BD. Los autores presentan una aplicación (Sistema Web) que opera con información geográfica y medioambiental.

Nnadi et al. [10] proponen una Infraestructura de Integración de BD's (DLII) ofreciendo un enfoque ligero y sistemático para integrar las colecciones y servicios de una BD. En este trabajo se se detalla como integrar las BD's por medio de metainformación. La principal contribución de esta investigación es proporcionar una infraestructura presumiblemente sencilla y sustentable para la integración de las colecciones y los servicios de BD.

Sin embargo, aunque existen propuestas de sistemas como el de Lima [9], para que puedan operar estos sistemas se requiere de ontologías por cada temática contenida en los documentos de las BD, lo que dificulta su uso con la información actual. Otros sistemas requieren de metadatos que relacionen la información como las propuestas de Seaman [8] y Nnadi [10], que similar al trabajo anterior requieren información adicional que se debe generar para poder utilizar sus propuestas.

Otros sistemas como OLIS [11] y NSDL, desarrollado a partir de la propuesta descrita en [10], también requieren recabar la información de otras BD's, integrarlas a sus RI's para posteriormente poder desplegar resultados de varias fuentes, lo que limita la flexibilidad para agregar nuevas fuentes.

3. Arquitectura del sistema

La Figura 1 muestra la arquitectura del sistema propuesto. Se presentan los participantes (tipos de agentes) y los componentes: BD's, servicios web y el registro de servicios. Dicha arquitectura ayudó a ofrecer una solución que responda la pregunta *¿Cómo hacer un sistema dinámico que permita incluir y modificar BD's sin tener que reprogramar el sistema?*, planteada en la Introducción.

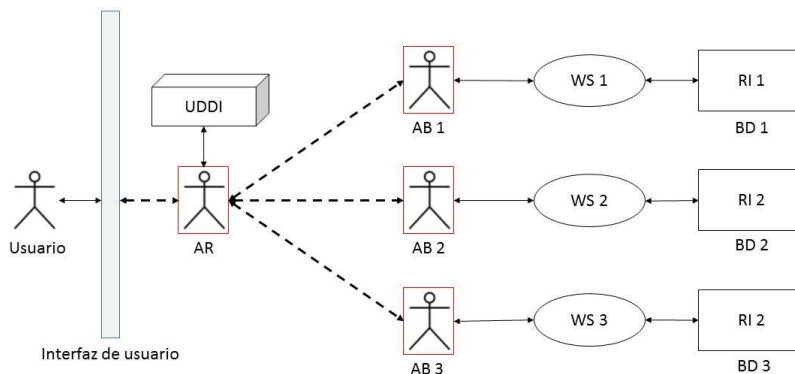


Fig. 1. Arquitectura general del sistema.

En el sistema contempla dos tipos de agentes o roles, los sistemas de BD y otros dos componentes.

- El Agente Representante (**AR**). El cual se encarga de la comunicación con el cliente (recibir la consulta y mostrarle los resultados), la búsqueda de servi-

cios (en el UDDI), así como de comunicarse con los AB_j . Mayor información se dará en la siguiente sección donde se observan las fases de la Integración.

- El Agente Bibliotecario (**AB**). Es el agente que se encarga de hacer las búsquedas en las BD's, a partir de la consulta hecha por el AR, enviar los resultados y ordenarlos de acuerdo a un determinado criterio.
- Los Servicios Web (**WS**). Son la interfaz a las BD's. Éstos permiten que los agentes o usuarios puedan consultar los sistemas de BD's (bajo diferentes criterios de búsqueda), ofrecen una interfaz estandarizada, reusable e interoperable (características propias de este tipo de sistemas).
- Otro de los componentes es el registro de servicios Web (**UDDI**). En éste se almacena dinámicamente información de servicios web como: descripción de lo que realizan, categoría, ubicación, interfaz de uso, así como los protocolos estándar que permiten comunicarse con dicho servicio.

Mayor detalle de la implementación de la arquitectura es mostrada en la subsección 5.2. En la siguiente sección se describen las fases de la composición de información, lo cual muestra el funcionamiento del SMA y como los Agentes cooperativamente resuelven el problema.

4. Fases de integración de las bibliotecas digitales

La construcción de la BD virtual se realiza por medio de 8 fases. Cada fase se describe principalmente por dos eventos: el inicio y la acción que se realiza. La figura 2 muestra la secuencia de las fases así como el agente encargado de la detonación.

Es importante resaltar que estas fases fueron propuestas para dar respuesta a la pregunta *¿Cómo integrar la información de varias fuentes?* Específicamente las fases 5-7 proponen un método para dar solución a la pregunta *¿Cómo reordenar (ranking) información ya ordenada de varias fuentes de acuerdo a una consulta y criterio?*, preguntas que fueron previamente planteadas en la sección 1.

Para que el sistema funcione se asume que se tiene un AR y un número n de AB 's, donde $n \geq 1$, y se representa a un AB cualquiera como AB_j , donde j puede ser un número entre 1 y n . Mayor detalle de la implementación de estas fases se puede encontrar en la subsección 5.3

Fase 1. Búsqueda de las bibliotecas digitales

Esta etapa se puede desplegar en dos momentos:

- 1) El primer momento es cuando se arranca el sistema, el AR consulta en el UDDI que BD's ofrecen sus servicios a través de WS y guarda una lista de ellos.
- 2) El segundo momento se presenta cuando el AR determina que es necesario consultar de nuevo el UDDI, debido a que no están disponibles algunas de las BD's o se requiere buscar más.

Fase 2. Recepción de la Consulta

Esta etapa se presenta cuando el usuario del sistema desea realizar una búsqueda. El AR obtiene una solicitud del usuario con la consulta y opcionalmente el criterio información que posteriormente le será enviada a cada AB_j .

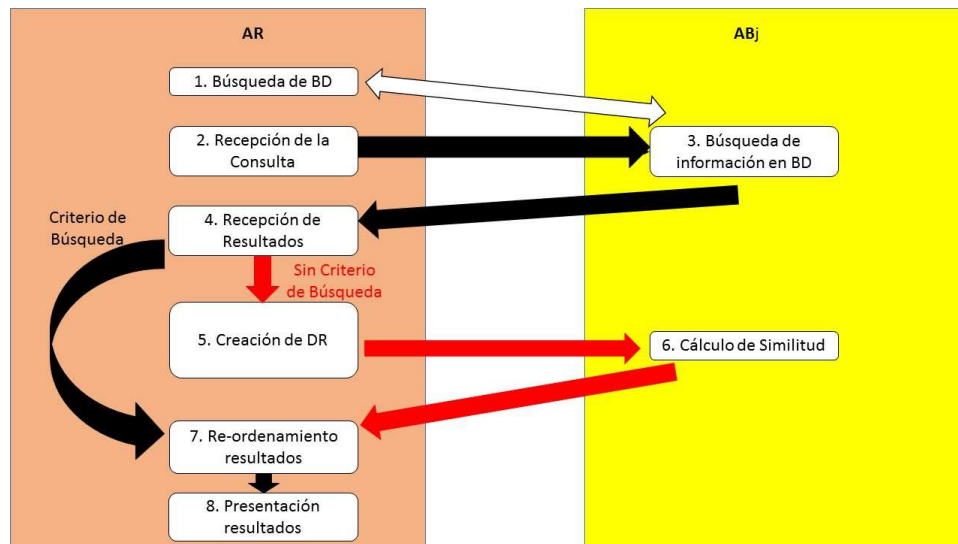


Fig. 2. Fases de Integración de BD's

Fase 3. Búsqueda de información en las bibliotecas digitales

Esta etapa comienza cuando el *AR* tiene una consulta del usuario. El agente *AR* invita a los *AB_j* a participar para resolver el problema. Cada *AB_j* selecciona una *BD* y manda su propuesta donde incluye la *BD* en la cual quiere buscar. Posteriormente, el *AB_j* busca documentos en la *BD* y envía los resultados a *AR*.

Fase 4. Recepción de resultados de las bibliotecas digitales

El *AR*, resultado de la comunicación con el *AB_j*, recibe los resultados de la búsqueda. Si se dio en la solicitud algún criterio (fecha o relevancia de la publicación) se pasa directamente a la fase 7, ya que la fecha o el número de referencias (relevancia) servirán para ordenar los documentos. En el caso de no haberse definido ningún criterio se continúa con la etapa 4, la cual se describe a continuación.

Fase 5. Creación del documento de referencia

Si en la solicitud no se especificó algún criterio de búsqueda entonces el *AR* hace una expansión de la consulta con la finalidad de proponer un Documento de Referencia (**DR**). Cada *AB_j* utilizará el **DR** para ordenar sus resultados, de acuerdo a la similitud. Para formar el documento de referencia el *AR* puede usar su conocimiento (por ejemplo, información que haya obtenido de las preferencias del usuario o de los temas), y el conocimiento de cada agente *AB_j*.

Fase 6. Cálculo de la similitud de los resultados con respecto al documento referencia

Una vez que el *AB* tiene el **DR**, éste lo compara contra todos los resúmenes. Los agentes se ponen de acuerdo para determinar un criterio de comparación.

Fase 7. Reordenamiento de resultados

Una vez que el *AR* recibe los resultados ordenados de algún AB_j , entonces el *AR* tomando en cuenta el criterio de comparación ordena los resultados.

Fase 8. Presentación de resultados

Los resultados son presentados al usuario de manera ordenada de acuerdo al criterio (o similitud).

5. El sistema integrador de bibliotecas digitales

En esta sección se presentan la interfaz del sistema y mayor detalle de la implementación de la arquitectura y las fases descritas en las secciones 3 y 4 respectivamente.

5.1. Interfaz del sistema

El SMA Integrador de BD's es accedido a través de una Sistema Web. Se usa de manera similar a los buscadores de información en Internet y BD's, escribiendo las palabras clave y presionando el botón de búsqueda.

En el caso de que se desee utilizar el criterio de búsqueda “año de la publicación” basta con poner antes o después de las palabras clave *criteria:year*. De manera similar se puede seleccionar el criterio de “relevancia de los artículos” *criteria:relevance*. Si hay resultados de la búsqueda estos son presentados de manera ordena (Ver Figura 3).



Fig. 3. Resultados de una búsqueda en el SMA integrador de BD's

5.2. Componentes utilizados en la arquitectura

Con la finalidad de crear y probar el prototipo del Sistema Multiagente Integrador de Bibliotecas Digitales se consideraron tres diferentes BD's: Springer

Link, SciELO e IEEE Xplorer. La selección de estas tres bibliotecas digitales se realizó debido a que son muy consultadas por la comunidad científica, además de que técnicamente permiten hacer consultas por medio del protocolo GET, es decir permiten enviar las consultas en el URL facilitando la integración.

En relación a los componentes, se programaron tres Servicios Web (WS), uno para cada BD, que funcionan como interfaz para que puedan utilizarlos los agentes que integran el sistema. Gran parte del sistema fue desarrollado en lenguaje de programación Java, ya que ofrece herramientas que facilitan el desarrollo de los servicios web y sus clientes, además de que el UDDI utilizado fue el jUDDI, el cual cuenta con un API en Java para que pueda ser utilizado mediante su código.

Con respecto a los participantes, se programaron un AR y tres AB_j (uno por cada BD).

5.3. Detalles de la implementación de las fases

Con la finalidad de facilitar la comunicación entre los diferentes actores se programaron diferentes protocolos de interacción basados en el estándar de FIPA [12]. A continuación se da mayor información sobre la implementación de cada fase.

Con respecto a la **Fase 1**, para encontrar todas las fuentes de documentos disponibles, el AR consulta el UDDI, donde solo hay registrados WS de BD's. Todos los servicios que AR encuentra los agrega a su lista para posteriormente enviarlos a los AB_j 's. Esta consulta la realiza cuando se arranca el sistema o cuando un agente AB_j falla al tratar de consumir al WS de la BD que le corresponde. En ese caso AB_j comienza el protocolo de interacción FIPA-Request-When, la condición es que el WS que le corresponde no es localizado, no contesta, o no envía los resultados en un formato que pueda procesar el agente. Como resultado del protocolo de interacción el agente AB_j esperaría una nueva asignación de WS el cual consultar.

En la **Fase 2**, el usuario, a través de la interfaz del sistema, activa el protocolo de comunicación FIPA-Request, enviando una solicitud de búsqueda de documentos en bibliotecas digitales al AR. En el mensaje de solicitud (request) se le envía al agente, la consulta y opcionalmente el criterio de búsqueda (fecha ó relevancia de la publicación). La comunicación entre el AR y el usuario terminará si el agente se niega, falla o reporta los resultados.

La **Fase 3** comienza cuando el AR arranca el protocolo de interacción FIPA-Iterated-Contract-Net. Envía una llamada a participar a los AB_j 's, en la llamada incluye las condiciones de ejecución, la consulta, el criterio y la lista de posibles BD's en las cuales puede buscar. El AR puede estar revisando las propuestas con la finalidad de hacer búsquedas en la mayoría de las BD's. Cada AB_j tiene un cliente de WS para consultar al WS de la BD, por el momento todas las interfaces de los WS son iguales.

El AR almacena, en la **Fase 4**, temporalmente los resultados de las búsquedas. En el caso de que se haya escogido algún criterio de búsqueda pasa a la fase 7, de lo contrario se continua con la siguiente fase.

En la **Fase 5**, debido a que en el sistema no se considera el conocimiento de *AR* para crear el DR, solo se toma en cuenta el primer resultado que regresa cada AB_j asumiendo que es el que mejor satisface la consulta. Luego entonces, el DR se crea por medio de la unión de los términos contenidos en los resúmenes de los artículos (retroalimentación ciega) que fueron obtenidos como el primer resultado de cada BD.

Posteriormente dicho documento se pre-procesa (DRP): eliminando palabras cerradas y signos de puntuación, transformado todas las letras a minúsculas, y utilizado un lematizador. Una vez que se cuente con el DRP éste se envía a cada AB_j .

En la **Fase 6**, AB_j compara los resúmenes de los documentos encontrados contra el DRP de la siguiente manera:

Se preprocesan todos los resúmenes de la colección r_j , donde el índice j representa al AB que lo encontró. Se obtiene un nuevo documento rp_j : eliminando palabras cerradas, signos de puntuación y se transforman todas las letras a minúsculas para que finalmente se utilice un lematizador.

Una vez teniendo todos los rp_j se calcula la similitud (coeficiente Jaccard) $sim(DRP, rp_j) = \frac{T_{DRP \cap T_{rp_j}}}{T_{DRP \cup T_{rp_j}}}$. Es decir cantidad de elementos que contiene la intersección de los conjuntos de términos de ambos documentos entre la cantidad de términos contenidos en la unión. Calculadas todas las similitudes estas se mandan junto con los índices de los documentos re-ordenados al agente *AR*.

Para acomodar los resultados en la **Fase 7**, el *AR* utiliza el ordenamiento por inserción, tomando en cuenta las similitudes calculadas por los AB_j , con respecto al DR.

En la última fase, **Fase 8**, se presentan todos los documentos encontrados por los AB 's y por cada documento encontrado se muestra: el nombre del artículo, un fragmento del resumen y la liga al sitio de la BD donde se puede consultar el artículo completo.

6. Pruebas

Las pruebas que se realizaron con respecto a la arquitectura fueron las siguientes:

- Se probó el sistema con 1 a 5 AB_j , debido a que hasta al momento solo hay disponibles (a través de WS) tres Bibliotecas Digitales, el sistema con 3 a 5 Agentes no mostró diferencia con respecto los resultados considerados (tomando en cuenta las 3 BD's) y el tiempo de respuesta del sistema fue de 2 a 3 minutos para un aproximado de 75 resultados.
- Se agregaron y eliminaron los WS de las BD's del UDDI para probar el dinamismo del sistema, se examinó con una, dos y tres WS de BD's y no se detectó ningún problema con respecto a la operación del sistema. El *AR* operó sobre las BD's que se encontraban registradas (una, dos o tres) mostrando los resultados del mismo número de BD's.

- Con respecto a la Integración de Información y Ordenamiento de Resultados de Diferentes fuentes se diseñó un pequeño experimento para probar dos formas de integración. Se realizó la búsqueda en las tres BD's con la consulta "Web Services and MultiAgent System Integration", recuperándose 70 resultados.

Posteriormente, con la finalidad de evaluar la fusión de documentos se hizo un etiquetado manual de los resultados marcando solo aquellos que hablaban sobre la integración de estas tecnologías, de este etiquetado solo 37 resultados fueron relevantes.

Los algoritmos probados para fusionar documentos fueron: como *Gold Standard* tomamos la integración Round-Robin (efecto *Skimming*) la cual consta en ingresar los resultados de las 3 BD en 3 pilas, los elementos se ingresan del resultados con peor *ranking* al mejor, quedando en el tope de la pila el mejor resultado. Se toma un elemento de cada pila intercalándolos hasta quedar vacíos.

El segundo algoritmo probado fue el descrito en la subsección 5.3 para las fases 5-7. Se tomó el mejor resultado de cada BD para la consulta previamente mencionada. Teniendo los tres resultados y mediante la retroalimentación ciega se logró extender la consulta con conjuntos de términos relevantes como: "service oriented computing", "loose coupled", "infraestructure", "distributed applications", "autonomic computing", "service discovery", "service description", "service invocation", "dynamic integration", "seamless integration", "service interoperability", "FIPA", "service oriented architecture".

Se obtuvo el DR y se pre-procesó para obtener el DRP. Se pre-procesaron todos los resúmenes de los resultados obtenidos de las BD, y cada AB_j cálculo la similitud (coeficiente Jaccard) de sus documentos encontrados con respecto a DRP. Cada AB_j envió las similitudes al AR quien se encargó de ordenarlos (por inserción) de acuerdo a su similitud.

Se compararon los resultados de las fusiones obtenidas de los dos algoritmos, siendo algoritmo propuesto el que arrojó mejor Precisión y Recuerdo (Ver Figura 4), ordenando la mayoría de documentos relevantes a la consulta en los primeros documentos mostrados al usuario. El mejor desempeño del algoritmo propuesto se alcanzó con los primeros 40 resultados ($P@40$) con $P=.725$ $R=.78$ contra $P=.575$ $R=.6216$ del primer método, razón por la cual se seleccionó para implementarlo en el sistema.

7. Conclusiones y trabajo futuro

Consideramos que la combinación de las tecnologías de SMA y SOA son una buena opción para generar sistemas de integración de información, ya que los agentes se pueden organizar para conseguir y ordenar la información. Con lo que respecta a la integración de información, proveniente de RI's, los algoritmos de integración y ordenamiento utilizados mostraron, para la mayoría de las pruebas, resultados aceptables.

Debido a que este trabajo enmarca el principio de la investigación y desarrollo del sistema se omitieron otras pruebas, sin embargo se ha planteado incluir otros

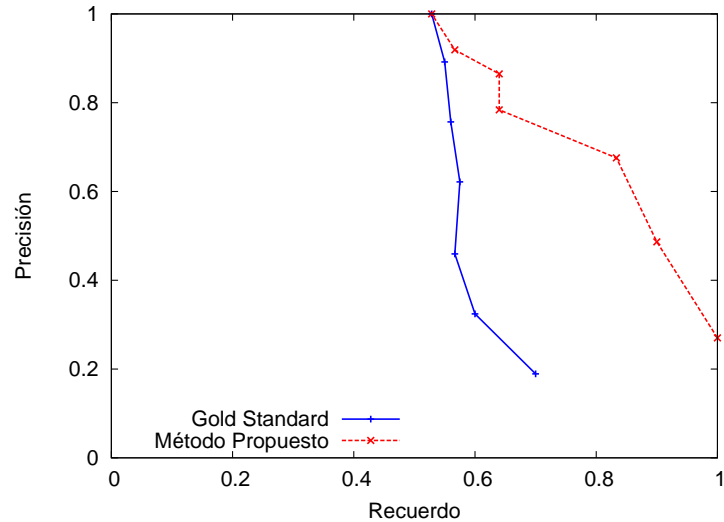


Fig. 4. Resultados (P y R) de los integradores Round Robin y método propuesto

protocolos de interacción entre agentes, evaluar comportamiento con un mayor número de agentes, consultas y BD's.

Así mismo, se ha planteado probar otros métodos de integración y ordenamiento de información. Otra de las tareas futuras es integrar a la tecnología de la Web Semántica, para desarrollar descripciones de servicios semánticos y mecanismos para que los agentes puedan realizar búsquedas más eficientes.

Así mismo, se propone dotar de herramientas al agente para extender consultas en base a su conocimiento del perfil del usuario.

Agradecimientos. Agradecemos a la Universidad Autónoma Metropolitana (UAM) y al proyecto, auspiciado por el Consejo Nacional de Ciencia y Tecnología (CONACYT), número CB2010/153315 por el apoyo para la asistencia a este evento. También queremos hacer un reconocimiento especial a la alumna de la UAM-Cuajimalpa Ruth Esquivel Pérez por haber realizado la programación de algunos módulos del sistema aquí descrito, como parte de su Proyecto Terminal

Referencias

1. New Jersey Institute of Technology, Project: Service Integration of Digital Libraries. <http://nsdl.org/about> (Última visita en Diciembre de 2014)
2. Newcomer, E., Lomow, G.: Understanding SOA with Web Services. Addison Wesley (2005)

3. Durfee, V.R., Lesser, E.H., Corkill, D.D.: Trends in Cooperative Distributed Problem Solving In: IEEE Transactions on Knowledge and Data Engineering, Vol. March 1989, pp.63–83, IEEE (1989)
4. Sycara, K., Paolucci, M., Ankolekar, A., Srinivasan, N.: Automated Discovery, Interaction and Composition of Semantic Web Services Journal of Web Semantics, Vol. 1, No. 1, pp. 27–46, Elsevier (2003)
5. Vogt C. C. and Cottrell G. W.: Predicting the performance of linearly combined ir systems. In: 21st International Conference on Research and Development in Information Retrieval, Melbourne, Australia (1998)
6. <http://www.greenstone.org> (Última visita en Diciembre de 2014)
7. Ibrahim,I. K., Schwinger, W.: Data Integration in Digital Libraries: Approaches and Challenges. In: Software Competence Center Hagenberg, Austria (2001)
8. Seaman, D.: Aggregation, Integration and Openness: Current Trends in Digital Libraries. In Proc. Digital Libraries and Knowledge Communities in Networked Information Society, Tsukuba, Ibaraki, Japan, pp.105–110 (2004)
9. Lima, T., Sheth, A., Ashish,N., Guntamadugu, M., Lakshminarayan, S., Palse-na,N., Singh,D.: Digital Library Services Supporting Information Integration over the Web. In: Workshop on Information Integration on the Web, pp. 19–26 (2001)
10. Nnadi, N., Bieber, M.: Towards Lightweight Digital Libraries Integration. (2004)
11. <http://www.bodleian.ox.ac.uk/bdlss/olis-ils> (Última visita en Diciembre de 2014)
12. <http://www.fipa.org/> (Última visita en Diciembre de 2014)

Detección de patrones de movilidad en datos de localización aplicando modelos de tópicos

Manuel Arturo Suárez-Améndola¹, Salvador Ruiz-Correa², Rogelio Hasimoto-Beltrán³

¹ Instituto Tecnológico Superior de Escárcega, México

² Center for Mobile Life Studies, México

³ Centro de Investigación en Matemáticas A.C., México

masuareb@hotmail.com, src@cmls.pw, hasimoto@cimat.mx

Resumen. De manera natural las personas vamos formando rutinas para adaptarnos al entorno social en el cuál nos desempeñamos. Estas rutinas se ven reflejadas en las secuencias de ubicaciones que visitamos para desplazarnos hacia los lugares donde realizamos nuestras actividades diarias, sin embargo, ¿cómo podemos determinar la presencia de ellas? El presente trabajo consiste en un análisis exploratorio de un conjunto de datos de movilidad de usuarios utilizando modelos de tópicos con el fin de determinar si existe una estructura subyacente en la movilidad de las personas de acuerdo a las rutinas que éstas manifiestan en sus vidas. Se presenta la introducción al problema explicando el modelo a utilizar así como su desarrollo teórico, validación experimental de su funcionamiento a través de conjuntos de datos sintéticos y su aplicación a un conjunto de datos real con presentación de los resultados obtenidos y conclusiones.

Palabras clave: reconocimiento de patrones, análisis exploratorio, datos móviles, modelos de tópicos.

1. Introducción

Es interesante observar cómo las personas van formando hábitos en sus vidas generando rutinas para la mayoría de sus actividades a través de la continua repetición de las mismas. Si bien, el comportamiento humano es un aspecto muy difícil de analizar, intuitivamente sabemos que las personas definen regiones de permanencia y secuencias de ubicaciones que visitan basados en los roles que como persona desempeñan dentro de una sociedad. Aunque el ser humano tiene el potencial de exhibir patrones de comportamiento aleatorio, existen ciertas rutinas fácilmente identificables en el contexto de su quehacer diario; por ejemplo, consideremos los pasos típicos de una secuencia de ubicaciones para una persona en un día laboral:

1. Levantarse a una hora definida.
2. Salir de su casa y tomar transporte (ya sea privado ó público).

3. Permanecer en su lugar de trabajo un período de tiempo definido.
4. Salir de la oficina y tomar transporte.
5. Regresar a casa.

Esta serie de ubicaciones en ese orden específico sugieren la presencia de patrones de comportamiento en los seres humanos en las ubicaciones que visita con base en sus rutinas de comportamiento en su vida diaria.

1.1. Modelos de tópicos

Los modelos de tópicos se desarrollaron inicialmente para el análisis de grandes colecciones de documentos de texto [6,12]; de manera simultánea e independiente, modelos similares se aplicaron en el área de biología, específicamente, en genética, donde los modelos de tópicos se han aplicado para la identificación y clasificación de poblaciones de genotipos [16]. Actualmente se han aplicado satisfactoriamente en otros contextos tales como ubicación [8] y proximidad física [2,7]. La capacidad de los modelos para extraer la información semántica y contextual de un documento los convierten en una herramienta fundamental para el análisis de texto, tomemos por ejemplo, la Enciclopedia Libre *Wikipedia*⁴ [1], la cuál es un compendio de más de 4 millones de artículos en Inglés (al 10 de Abril de 2015 según la página principal) clasificados en varios temas principales (Artes, Historia, Sociedad, Matemáticas, etc.). De manera natural, existen artículos que pertenecen a más de una clasificación por su contenido temático, por lo cuál podemos preguntarnos, ¿existen clasificaciones que no se hayan considerado entre las actuales?, ¿hay clasificaciones a las que un documento pertenezca que no se hayan detectado?. Las anteriores y otras preguntas surgen al imaginar la extensión del conocimiento plasmado en un número tan grande de artículos y que continuamente va creciendo.



Fig. 1. Buscador por Tópicos de Wikipedia

⁴ www.wikipedia.org

*Wikipedia Topics*⁵ [4] es un ejemplo de la aplicación de un modelo de tópicos a un conjunto de artículos de la *Wikipedia*. En la Figura 1 podemos ver la organización de los tópicos estimados a partir de la aplicación del modelo en el conjunto de documentos y en la Figura 2 podemos ver el conjunto de documentos relacionados a un tópico. Podemos ver que los Modelos de Tópicos nos permiten inferir de manera automática un contenido semántico implícito en un conjunto de documentos de acuerdo al contenido temático de cada uno de ellos.

{household, population, female}		
words	related documents	related topics
household	Census	{township, household, population}
population	Race (classification of human beings)	{household, population, family}
female	Ramona, California	{city, population, household}
family	The Bellmores, New York	{county, mile, population}
median	Chester, Connecticut	{utc_offset, utc_offset_dst, timezone}
income	Deep River, Connecticut	{line, north, south}
average	Woodbury, Connecticut	{borough, population, unit_pref}
year	New Hartford, Connecticut	{area, community, home}
km ²	Westbrook, Connecticut	{town, population, incorporate}

Fig. 2. Artículos relacionados a un tópico

Por su capacidad de análisis sobre conjuntos de datos discretos, los modelos de tópicos se vuelven una herramienta muy útil para el reconocimiento de patrones. Por lo tanto, el interés principal de éste trabajo es la identificación de patrones estructurales en los conjuntos de datos de ubicaciones geográficas obtenidas con dispositivos móviles aplicando modelos de tópicos.

1.2. Estructura del trabajo realizado

En este artículo se analiza un modelo de tópicos aplicado a un conjunto de datos que representan secuencias de ubicaciones geográficas con el objetivo de identificar una estructura presente en los datos que representen patrones de comportamiento en las personas. Se analizan aspectos teóricos del modelo así como una verificación experimental del funcionamiento del mismo utilizando datos sintéticos generados específicamente para probar aspectos del funcionamiento del modelo. Finalmente se aplica el modelo a un conjunto de datos de movilidad de usuarios observando los resultados obtenidos así como las conclusiones y trabajo a futuro.

2. Distant N-Gram Topic Model (DNTM)

Dos de los aspectos mas relevantes para el análisis de secuencias de datos discretos son la capacidad del modelo para expresar la pertenencia de un elemento

⁵ <http://www.princeton.edu/~achaney/tmve/wiki100k/browse/topic-presence.html>

a más de una categoría de clasificación y la capacidad para incluir información de dependencia entre los elementos en las secuencias de datos. Estas ventajas las presenta el modelo DNTM, el cual se describe brevemente en la sección 2.3 y a detalle en la sección 3.

2.1. Modelos de pertenencia mixta

$$p(v) = \sum_{h=1}^H p(v | h)p(h) \quad (1)$$

Un modelo mixto es uno en el cuál un conjunto de modelos es combinado para producir un modelo más expresivo [3]. En un modelo mixto, la variable v es visible y la variable h la indexa conformando los grupos a los cuáles puede pertenecer la variable v . Específicamente, podemos observar que una palabra puede pertenecer a más de un tópico lo cuál se indica con la medida de probabilidad asignada a ella en cada uno de los tópicos. Sin embargo, en un modelo mixto tradicional, un elemento sólo pertenece a un único grupo, por ejemplo en el algoritmo K-Means [14], un algoritmo que geoméricamente descubre K grupos de elementos dada una medida de distancia. En K-Means cada elemento del conjunto de datos se asocia a un grupo lo cuál se conoce como *hard-clustering*. A diferencia, el *soft-clustering* considera que un elemento puede pertenecer a más de un grupo, de acuerdo a cierta medida de similitud. Los modelos de tópicos como LDA [6] y PLSI [12] consideran que un elemento de datos puede pertenecer a más de una clasificación (tópicos) de acuerdo a cierta medida de probabilidad.

2.2. Modelado de las dependencias

De manera natural, las personas inician sus rutinas diarias en un lugar en específico a partir del cuál se van moviendo a lo largo de sus rutinas definidas. Por lo tanto, es deseable que el modelo probabilístico a aplicar tenga ésta consideración en cuenta agregando dichas dependencias para poder capturar de manera correcta el significado semántico de su comportamiento. Sin embargo, se requiere evitar la explosión paramétrica que ocurriría si se modelan de manera explícita las dependencias entre cada uno de los elementos de la secuencia. Por tanto, en [10] se sugiere el modelo expresado en la Figura 3, donde se puede apreciar que, en primer lugar, se define el modelo con base en secuencias de ubicaciones $q = (w_1, w_2, \dots, w_N)$ en vez de ubicaciones individuales, y en segundo lugar, que se define una dependencia a la primera ubicación visitada en la secuencia.

2.3. Descripción del modelo

El modelo DNTM [10] es un modelo derivado de LDA [5] el cuál es un modelo generativo bajo el cuál cada documento en una colección de texto (corpus) es representado con una distribución multinomial de tópicos los cuáles capturan el

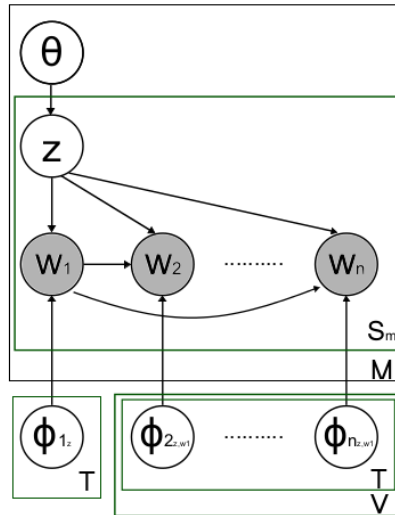


Fig. 3. Modelo Gráfico del modelo DNTM [10]

significado semántico de cada documento. A su vez, cada tópico es representado con una distribución multinomial sobre las palabras de un diccionario las cuáles capturan el significado semántico del tópico. En la modelación de patrones estructurales de movilidad de las personas, se considera una ubicación geográfica como una analogía a una palabra dentro de un documento y un intervalo de tiempo en el cuál se han dado las ubicaciones como una analogía a un documento. Es decir, se tiene un conjunto de elementos (ubicaciones geográficas) que pertenecen a diferentes clasificaciones (rutinas) de acuerdo a la sucesión de ellas. De ésta manera, el modelo de tópicos intenta descubrir o capturar el significado semántico de las secuencias de ubicaciones que un usuario visita con base en sus rutinas de movilidad.

3. Desarrollo del modelo

Como se observa en la Figura 3, θ y ϕ son las realizaciones Dirichlet de las distribuciones de los tópicos sobre los documentos (distribuciones multinomiales) las cuáles influyen directamente en la generación de la variable z (los tópicos latentes) y las variables w_n las cuáles dependen del tópico generado z , las variables ϕ_n y, para $j > 1$ del primer elemento w_1 . Los parámetros del modelo se muestran en el cuadro 1.

3.1. Estimación de parámetros

La estimación exacta de parámetros para el modelo DNTM como en el modelo LDA es intratable, por lo que se han desarrollado métodos de apro-

N	Longitud de la secuencia.
q	Una secuencia de N etiquetas consecutivas (w_1, \dots, w_N) .
m	Una instancia de un documento.
S_m	El número total de secuencias q en el documento m .
M	El número total de documentos en el corpus.
T	El número de tópicos latentes.
z	Una instancia de un tópico.
V	El tamaño del vocabulario.
Θ	La distribución de los tópicos dados los documentos.
Φ	La distribución de las secuencias dados los tópicos, donde $\Phi = \{\Phi_{1z}, \Phi_{2z, w_1}, \dots, \Phi_{nz, w_1}\}$
Φ_{1z}	La distribución de las etiquetas w_1 dados los tópicos.
Φ_{jz, w_1}	La distribución de las etiquetas w_j dada la primera etiqueta w_1 y los tópicos.

Tabla 1. Descripción de símbolos de los componentes del modelo DNTM

ximación basados en inferencia variacional [6] y simulaciones de Monte Carlo (MCMC) [11]. En [9] y [10] se desarrolla un método de inferencia por Monte Carlo utilizando un esquema de Gibbs Sampling colapsado para estimar los parámetros del modelo θ y ϕ a partir de la simulación de la distribución condicional $p(z | q)$, por lo que se requiere de la expresión $p(z_i | z_{-i}, q)$ donde z_{-i} denota la probabilidad condicional para la simulación de la distribución de un sólo tópico z_i condicionado a las observaciones q y los tópicos restantes z_{-i} .

3.2. Desarrollo del modelo (colapsado de variables)

A partir de la Figura 3 y bajo la asunción de *i.i.d.* presente en los datos, determinamos la expresión de la verosimilitud para desarrollar el colapsado de variables (marginalización de las variables θ y ϕ para determinar la probabilidad $p(z, w_{1:n})$). Adoptamos la siguiente notación para escribir la expresión de la verosimilitud:

1. Z el conjunto de tópicos (variables latentes) z_i .
2. W el conjunto de observaciones (secuencias) $w_{1:n}$.
3. Θ el conjunto de realizaciones Dirichlet de las distribuciones de tópicos sobre documentos θ .
4. Φ el conjunto de realizaciones Dirichlet de las distribuciones de palabras sobre tópicos ϕ .

$$\begin{aligned}
 P(Z, W, \Theta, \Phi | \alpha, \beta_{1:n}) = & \\
 & \prod_{m=1}^M \left[p(\theta_m | \alpha) \prod_{s=1}^S \left[p(z_{ms} | \theta_m) p(w_{1ms} | z_{ms}, \varphi_1) \prod_{j=1}^n p(w_{jms} | w_{1ms}, z_{ms}, \varphi_{jw_1}) \right] \right] \\
 & \prod_{t=1}^T \left[p(\varphi_{1t} | \beta_1) \prod_{j=2}^n \prod_{v=1}^V p(\varphi_{jtv} | \beta_j) \right] \quad (2)
 \end{aligned}$$

A partir de la expresión de la verosimilitud (2), se procede a integrar sobre los parámetros θ y ϕ para poder obtener la distribución requerida para el procedimiento de inferencia $p(z|q)$. Definiendo variables para la composición de los

productos en sumatorias sobre los exponentes de las distribuciones de probabilidad ν_{mt} , τ_{tv} y $\xi_{jw_1w_2}$, procedemos a la normalización de las distribuciones Dirichlet con el fin de realizar su integración la cuál es igual a 1. (La integración de una distribución de probabilidad sobre todo su dominio es igual a 1), lo que resulta en la distribución de probabilidad conjunta de Z y W .

$$\begin{aligned}
 P(Z, W | \alpha, \beta_{1:n}) = & \\
 & \prod_{m=1}^M \frac{\Gamma(\alpha T)}{\Gamma(\alpha)^T} \frac{\prod_{t=1}^T \Gamma(\nu_{mt} + \alpha)}{\Gamma(\sum_{t=1}^T \nu_{mt} + \alpha)} \\
 & \prod_{t=1}^T \frac{\Gamma(\beta_1 V)}{\Gamma(\beta_1)^V} \frac{\prod_{v=1}^V \Gamma(\tau_{tv} + \beta_1)}{\Gamma(\sum_{v=1}^V \tau_{tv} + \beta_1)} \\
 & \prod_{t=1}^T \prod_{j=2}^n \frac{\Gamma(\beta_j)^V}{\Gamma(\beta_j)^{V^2}} \frac{\prod_{w_1=1}^V \prod_{w_2=1}^V \Gamma(\xi_{jtw_1w_2} + \beta_j)}{\Gamma(\sum_{w_1=1}^V \sum_{w_2=1}^V \xi_{jtw_1w_2} + \beta_j)} \tag{3}
 \end{aligned}$$

Recordemos que la inferencia sobre ésta distribución es intratable [11] por lo que recurrimos a métodos aproximados y para utilizar Gibbs Sampling, requerimos de la probabilidad condicional $p(z_i | z_{-i}, w)$, es decir, la expresión que corresponda a la probabilidad condicional de una tópico z dados los tópicos restantes z_{-i} y las observaciones w . Por tanto, observando que para una observación z_i en el modelo su valor es igual a 1, aplicamos las propiedades de la función Gamma para simplificar la expresión de la verosimilitud obteniendo las expresiones para las probabilidades condicionales requeridas para el muestreador Gibbs (4).

$$\begin{aligned}
 p(z_{ms} = k | z^{-ms}, w_{1:n}, \alpha, \beta_{1:n}) \propto & \\
 (\nu_{mt}^{-ms} + \alpha) \frac{\tau_{tv}^{-ms} + \beta_1}{\sum_{v=1}^V (\tau_{tv}^{-ms} + \beta_1)} \prod_{j=2}^n \frac{\xi_{jtw_1w_2}^{-sm} + \beta_j}{\sum_{w_1=1}^V \sum_{w_2=1}^V \xi_{jtw_1w_2}^{-sm} + \beta_j} \tag{4}
 \end{aligned}$$

4. Verificación del modelo

4.1. Estructura de los datos sintéticos

Conjunto de Datos Sintéticos No. 1. El primer conjunto de datos se realiza con un vocabulario de 10 etiquetas distribuidas en 5 tópicos cada uno representado como una secuencia de 6 ubicaciones. Se genera un documento con 2,000 secuencias aleatorias asumiendo la misma probabilidad para todos los tópicos dentro de un mismo documento.

En la Figura (4) se muestran las distribuciones de probabilidad correspondientes a cada uno de los 5 tópicos, donde cada tópico contiene una secuencia de 6 posibles ubicaciones (eje x); la primera ubicación sobre el eje x corresponde a la etiqueta que se genera en primera posición de las secuencias y las ubicaciones posteriores sobre el mismo eje corresponden a las probabilidades de las etiquetas siguientes. El eje y corresponde a las posibles etiquetas del vocabulario ($V = 10$).

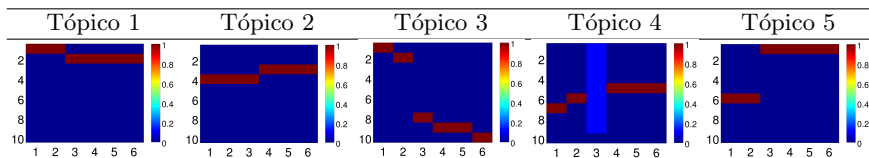


Fig. 4. Distribuciones de probabilidad de los tópicos del primer conjunto de datos sintéticos.

Obsérvase que en el tópicos 4 generamos en la tercera posición probabilidades similares para los elementos del vocabulario excepto el último.

Conjunto de Datos Sintéticos No. 2. La característica más relevante del conjunto de datos sintéticos No. 2 es la asignación de probabilidades para la generación de más de una etiqueta en la primera posición en las secuencias del documento, lo cuál se muestra en la figura 5.

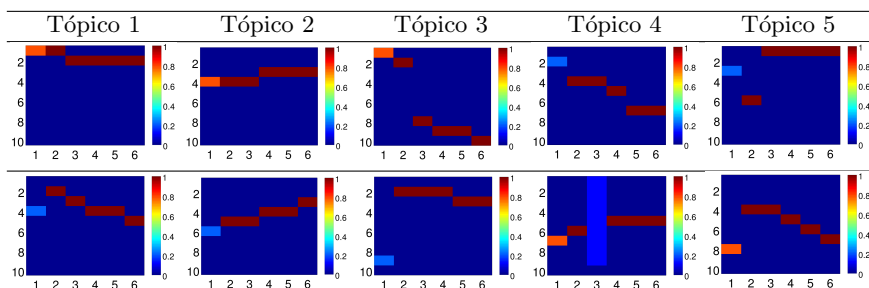


Fig. 5. Distribuciones de probabilidad de los tópicos del segundo conjunto de datos sintéticos.

Conjunto de Datos Sintéticos No. 3. Como podemos observar en la Figura 6, para el tercer conjunto de datos se generan 5 tópicos con distribuciones similares, donde sólo en el último elemento de las secuencias se observa una diferencia entre las etiquetas generadas. De ésta manera queremos determinar si el modelo es capaz de inferir diferencias entre estructuras de moviidades similares.

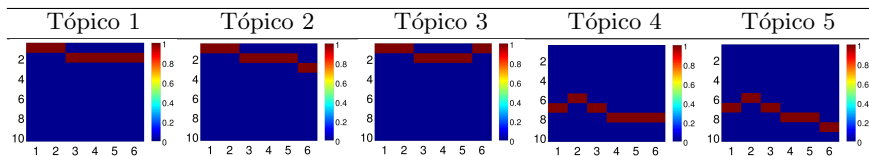


Fig. 6. Distribuciones de probabilidad de los tópicos del tercer conjunto de datos sintéticos.

Conjunto de Datos Sintéticos No. 4. En el último conjunto de datos deseamos verificar que el modelo sea capaz de identificar secuencias de longitudes diferentes aun a pesar de que él mismo no integre dentro de su estructura un mecanismo para determinar la longitud de cada secuencia. Se diseña entonces, un conjunto de 6 tópicos consistentes en secuencias de diferente longitud $N = 6$ y $N = 9$. En la Figura 7 se muestran las distribuciones de probabilidad para los tópicos, donde los tópicos 4 y 5 son los tópicos de longitud 9 y los restantes de longitud 6.

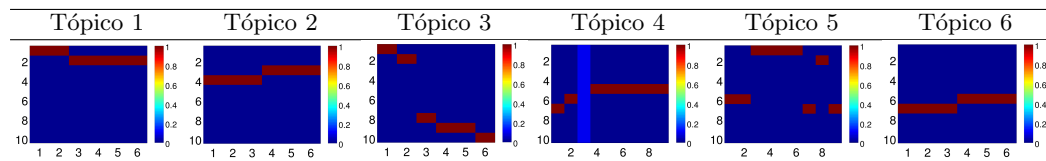


Fig. 7. Distribuciones de probabilidad de los tópicos del cuarto conjunto de datos sintéticos.

4.2. Resultados del modelo con datos sintéticos

Resultados del conjunto de Datos Sintéticos No. 1. Aplicando el modelo con los valores de $T = 5$ y $N = 6$, obtenemos en la Figura 8 que los resultados son bastante precisos para el conjunto de datos, observando que el tópicos 3 mezcló secuencias de tópicos diferentes y los tópicos 3 y 4 del modelo original se combinaron en el tópicos 4.

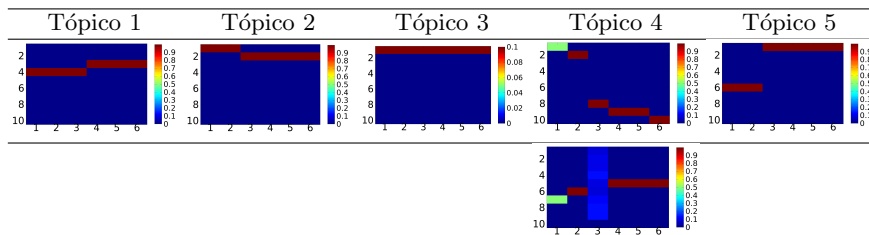


Fig. 8. Distribuciones de probabilidad obtenidas con el modelo con $N = 6$ y $T = 5$.

Resultados del conjunto de Datos Sintéticos No. 2. Podemos observar en la Figura 9 que los tópicos obtenidos son más precisos, sin embargo, uno de ellos no fue recuperado, lo cual puede indicar que es mejor considerar un número mayor de tópicos para permitir que el modelo pueda explorar el espacio de búsqueda de distribuciones de probabilidad que puedan corresponder a los parámetros que el modelo va a inferir sobre los datos.

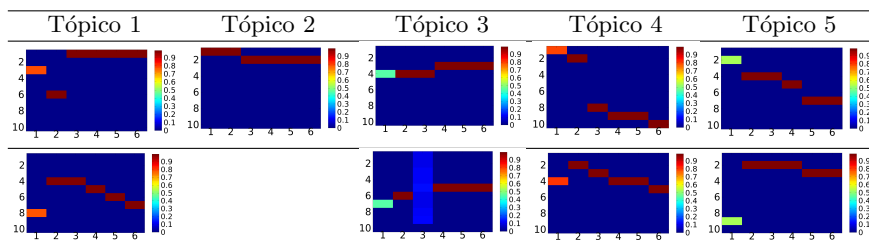


Fig. 9. Distribuciones de probabilidad obtenidas con el modelo con $N = 6$ y $T = 5$.

Resultados del conjunto de Datos Sintéticos No. 3. Podemos observar que el modelo no distingue de manera tan precisa entre los tópicos de estructura similar, como podemos apreciar en la Figura 10, los tópicos 1, 2, 4 y 5 tienen la misma estructura de secuencias difiriendo en la última pero combinando los tres tópicos de las distribuciones originales en los mismos tópicos. En el tópico 3 combinó los dos tópicos similares. Esto indica que el modelo es muy sensible a la presencia de estructuras de movilidad muy similares entre sí, no pudiendo distinguir pequeñas diferencias entre ellas.

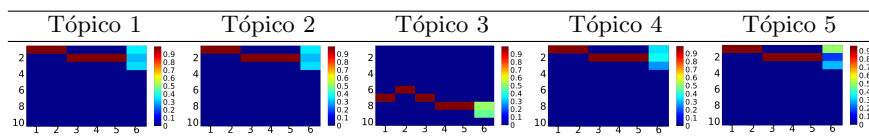


Fig. 10. Distribuciones de probabilidad obtenidas con el modelo con $N = 6$ y $T = 5$.

Resultados del conjunto de Datos Sintéticos No. 4. En las pruebas realizadas para éste conjunto de datos, se utilizaron valores de longitudes de secuencias $N = 6$, $N = 9$ y $N = 12$, observando que para $N = 6$ el modelo no alcanza a reconocer las estructuras de longitudes mayores dividiéndolo entre los tópicos detectados y para $N = 9$ el modelo reconoce de manera correcta longitudes menores pero las mayores las divide igualmente entre los tópicos. Mostramos entonces los resultados para $N = 12$ en la Figura 11 donde se muestran las distribuciones de máxima probabilidad para las secuencias detectadas.

5. Reconocimiento de patrones de movilidad en datos de ubicación geográfica

5.1. Datos Nokia: Lausanne data collection campaign [13]

El conjunto de datos móviles analizados consiste en el seguimiento de datos reales en la vida de 25 usuarios llevando un teléfono móvil Nokia modelo N95 en un periodo de nueve meses (del 01-10-2009 hasta el 01-07-2010).

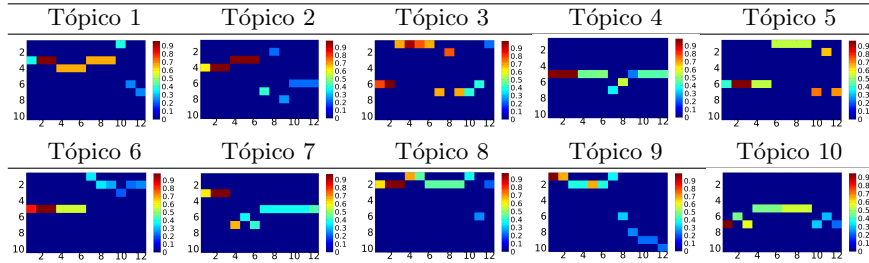


Fig. 11. Distribuciones de máxima probabilidad obtenidas con el modelo con $N = 12$ y $T = 10$.

En [10], obtienen los datos crudos de los teléfonos móviles y utilizan un algoritmo de extracción de datos [15] para el agrupamiento de las ubicaciones geográficas detectadas en un intervalo de tiempo basado en técnicas de agrupamientos de datos. Por tanto, se dispone de un conjunto de datos discreto multidimensional de tamaño $25 \times 335 \times 140$ (es decir, 25 usuarios, 335 días para cada uno de ellos y 140 mediciones de intervalos de 10 minutos para cada día).

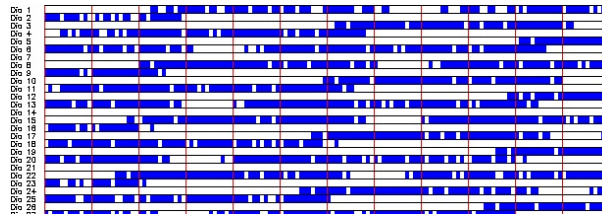


Fig. 12. Detección de ubicaciones geográficas correspondientes a un usuario

En la Figura 12 podemos observar una gráfica correspondiente a las detecciones de ubicaciones geográficas del usuario No. 1 del Conjunto de datos Nokia, en el eje x tenemos los 140 intervalos de tiempo correspondientes a 10 minutos cada uno de ellos, en el eje y tenemos los días durante los cuáles se hizo la detección y el color de cada celda corresponde a distintas ubicaciones pero donde el blanco se utiliza para indicar que no hubo detección correcta de ella.

5.2. Tópicos más probables

En la Figura 13 observamos un gráfico señalando los tópicos más probables para cada usuario, sobre el eje x tenemos cada uno de los 10 tópicos $T = 10$ y sobre el eje y tenemos a los usuarios, el color de cada celda indica la probabilidad de cada tópico para cada usuario. Observamos que en general, algunos tópicos definen la mayoría de la probabilidad de ocurrencia para cada usuario, teniendo algunos casos donde dicha probabilidad se distribuye de manera más uniforme.

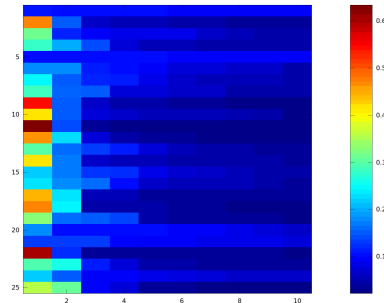


Fig. 13. Tópicos más probables obtenidos por el modelo DNTM sobre los 25 usuarios con 10 tópicos $T = 10$ y longitud de secuencia $N = 6$

5.3. Días más probables

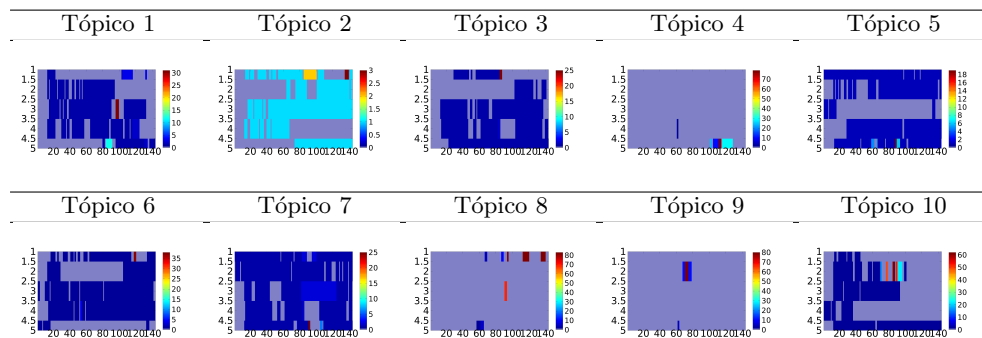


Fig. 14. Distribución de los 5 días más probables dados los tópicos.

En la Figura 14 podemos visualizar los 5 días mas probables para el usuario No. 1 con $T = 10$ y longitud de secuencia $N = 6$. Sobre el eje x se encuentran los intervalos de tiempo de 10 minutos definidos en el día (140) y sobre el eje y se encuentran los días más probables para cada tópico. El color de cada celda indica las diferentes etiquetas de las regiones de permanencia definidas para el usuario y las celdas en color gris indican que no hubo detección de ubicación en ese intervalo de tiempo.

5.4. Secuencias más probables

En la Figura 15 podemos visualizar algunos tópicos descubiertos por el modelo para el usuario No. 2, observamos la predominancia de las primeras etiquetas en los tópicos lo cuál proviene del hecho de que las etiquetas se ordenaron por

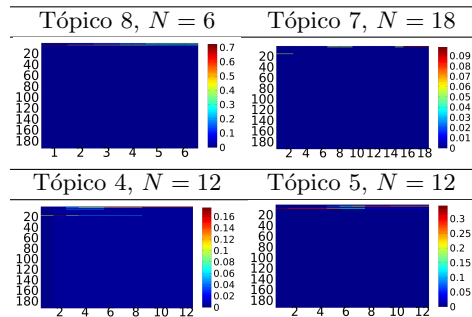


Fig. 15. Usuario 2, Distribución de Secuencias

frecuencia de aparición, aun así, observamos que la probabilidad se distribuye entre ellas. En los tópicos número 4 y 5 ($N = 12$) observamos también que las secuencias descubiertas por el modelo integran información de etiquetas diversas tal como en los datos sintéticos.

6. Conclusiones y trabajo a futuro

Los modelos de tópicos se han venido desarrollando desde sus primeras aplicaciones para modelado de texto y aplicaciones en genética, hasta dar el salto hacia su aplicación en el contexto de “reality mining”, lo cuál es una extensión natural debido a sus características inherentes al análisis de datos discretos y su potencial aplicación para discernir estructuras subyacentes en un conjunto de datos. Con base en los resultados obtenidos con los datos sintéticos y el conjunto de datos Nokia de movilidad se puede concluir que un modelo de tópicos es una herramienta adecuada para el reconocimiento de patrones en el conjunto discretizado de ubicaciones geográficas y que permiten la detección de estructuras de movilidad en las personas.

Durante el desarrollo del modelo se observaron varias áreas de oportunidad para continuar con el trabajo y mejorar los resultados, a continuación se presentan algunas sugerencias del trabajo que se pueden desarrollar:

- **Identificación de las longitudes de secuencia.**- Aplicar métodos no paramétricos para identificar de manera automática las longitudes de secuencia subyacentes en los datos.
- **Modelos alternativos de análisis.**- Aplicar extensiones al modelo probabilístico para identificar una mayor descomposición de la información, como el modelo de autor para identificar patrones comunes de movilidad en los usuarios (movilidad en grupos).
- **Mayor exploración del espacio de resultados.**- Implementar “Parallel Tempering” en el modelo para permitir la mayor exploración del espacio de resultados con el muestreador MCMC.

Referencias

1. Wikipedia la enciclopedia libre. <http://www.wikipedia.org/> (2015), [Fecha de consulta: 10-Abril-2015]
2. Bao, T., Cao, H., Chen, E., Tian, J., Xiong, H.: An unsupervised approach to modeling personalized contexts of mobile users. In: Data Mining (ICDM), 2010 IEEE 10th International Conference on. pp. 38–47 (Dec 2010)
3. Barber, D.: Bayesian Reasoning and Machine Learning. Cambridge University Press (2012)
4. Barlow, A.: Wikipedia topics. <http://www.princeton.edu/~achaney/tmve/wiki100k/browse/topic-presence.html/> (2015), [Fecha de consulta: 10-Abril-2015]
5. Blei, D.M.: Probabilistic topic models. Commun. ACM 55(4), 77–84 (Apr 2012), <http://doi.acm.org/10.1145/2133806.2133826>
6. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. J. Mach. Learn. Res. 3, 993–1022 (Mar 2003), <http://dl.acm.org/citation.cfm?id=944919.944937>
7. Do, T.M.T., Gatica-Perez, D.: Groupus: Smartphone proximity data and human interaction type mining. In: Proceedings of the 2011 15th Annual International Symposium on Wearable Computers. pp. 21–28. ISWC '11, IEEE Computer Society, Washington, DC, USA (2011), <http://dx.doi.org/10.1109/ISWC.2011.28>
8. Farrahi, K., Gatica-Perez, D.: Probabilistic mining of socio-geographic routines from mobile phone data. Selected Topics in Signal Processing, IEEE Journal of 4(4), 746–755 (Aug 2010)
9. Farrahi, K.: A Probabilistic Approach to Socio-Geographic Reality Mining. Ph.D. thesis, École Polytechnique Fédérale de Lausanne (2011)
10. Farrahi, K., Gatica-Perez, D.: A probabilistic approach to mining mobile phone data sequences. Personal and Ubiquitous Computing 18(1), 223–238 (2014), <http://dx.doi.org/10.1007/s00779-013-0640-8>
11. Griffiths, T.L., Steyvers, M.: Finding scientific topics. Proceedings of the National Academy of Sciences 101(suppl 1), 5228–5235 (2004), http://www.pnas.org/content/101/suppl_1/5228.abstract
12. Hofmann, T.: Probabilistic latent semantic analysis. In: In Proc. of Uncertainty in Artificial Intelligence, UAI'99. pp. 289–296 (1999)
13. Kiukkonen, N., J., B., Dousse, O., Gatica-Perez, D., J., L.: Towards rich mobile phone datasets: Lausanne data collection campaign. In: Proc. ACM Int. Conf. on Pervasive Services (ICPS), Berlin. (7 2010)
14. MacQueen, J.B.: Some methods for classification and analysis of multivariate observations. In: Cam, L.M.L., Neyman, J. (eds.) Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability. vol. 1, pp. 281–297. University of California Press (1967)
15. Montoliu, R., Gatica-Perez, D.: Discovering human places of interest from multimodal mobile phone data. In: Proceedings of the 9th International Conference on Mobile and Ubiquitous Multimedia. pp. 12:1–12:10. MUM '10, ACM, New York, NY, USA (2010), <http://doi.acm.org/10.1145/1899475.1899487>
16. Rosenberg, N.A., Pritchard, J.K., Weber, J.L., Cann, H.M., Kidd, K.K., Zhivotovsky, L.A., Feldman, M.W.: Genetic structure of human populations. Science 298(5602), 2381–2385 (2002), <http://www.sciencemag.org/content/298/5602/2381.abstract>

Análisis de deserción escolar con minería de datos

José Luis Aguirre Mendiola¹, Rosa María Valdovinos Rosas², Juan Alberto Antonio Velazquez^{1,3}, Roberto Alejo Eleuterio³, José Raymundo Marcial Romero²

¹ Universidad de Ixtlahuaca CUI, Ixtlahuaca de Rayón, Estado de México, México

² Universidad Autónoma del Estado de México, Facultad de Ingeniería, Estado de México, México

³ Tecnológico de Estudios Superiores de Jocotitlán, Estado de México, México

Resumen. En el presente trabajo se analiza una base de datos para identificar las causas de la deserción escolar en la carrera de ingeniería en computación de la Universidad de Ixtlahuaca CUI (UICUI) a través de técnicas de minería de datos. Para ello, se aplicaron reactivos para obtener información relacionada al semestre, número de cuenta, edad, estado civil, conocer si trabaja para solventar sus estudios o cuenta con el apoyo de sus padres y forma de elección de la carrera. Una vez finalizado el proceso de minería de datos fue posible identificar algunas causas de la deserción escolar en la UICUI.

Palabras clave: minería de datos, deserción escolar, arboles de decisión.

1. Introducción

La deserción estudiantil se da cuando un número determinado de estudiantes matriculados no siguen sus estudios universitarios, bien sea por abandono producido por la insatisfacción, por repetir semestres, por causas familiares, por el grado de complejidad, entre otras. Este problema, tiene efectos de tipo financiero, académico y social que implican la pérdida de esfuerzos y recursos en donde más de la mitad de los estudiantes que comienzan una carrera universitaria no terminan sus estudios [1].

En los últimos años, la carrera de Ingeniería en computación de la Universidad de Ixtlahuaca (CUI) ha mostrado alarmantes índices de bajas temporales y totales en los primeros semestres de la carrera, con una pobre tasa de egresados.

Las causas centrales viables de estudio son cuestiones tanto familiares como económicas, además de considerar el grado de dificultad que la carrera tiene, de acuerdo al grado del uso del razonamiento lógico y matemático.

Al respecto, la minería de datos (MD) [2] proporciona una alternativa de solución para el análisis de los fenómenos no explícitos en bases de datos. Es considerada una herramienta que permite analizar grandes bases de datos de forma más detallada, facilitando la toma de decisiones. En este sentido, el objetivo principal de la MD es integrar, analizar datos y extraer modelos que forman determinados patrones a partir de los datos analizados, con los que permite obtener una descripción del comportamiento de los datos, tendencias y correlaciones [3].

La Minería de Datos se constituye la etapa de descubrimiento en el proceso de KDD [4] el cual consiste en el uso de algoritmos concretos los cuales generan una enumeración de patrones a partir de los datos anteriormente procesados, apoyándose con algoritmos de Aprendizaje Automático [2].

El estudio aquí presentado, aplica la metodología de MD con la intención de identificar vulnerabilidades en el área escolar de Ingeniería en computación para que la carrera tenga más potencial y se desarrollen mejor los profesionistas dentro de esta área que es muy demandada actualmente.

2. Caso de estudio

La Universidad de Ixtlahuaca CUI lleva un transcurso de 38 años, la escuela fue fundada en el año de 1977 iniciando originalmente como la Preparatoria Regional “Químico José Donaciano Morales”, 15 años después en 1992 cuando se tuvo la necesidad de que los alumnos egresados de las escuelas preparatorias ingresaran a un nivel superior se inició la gestión para incorporar a la UAEM carreras universitarias. El 25 de Julio de 1996 el Consejo Universitario de la UAEM determinó incorporar los estudios de la licenciatura en Ingeniería en Computación y no fue sino hasta finales del 2011 que se otorga el nombre de Universidad de Ixtlahuaca CUI (<http://uicui.edu.mx/>).

En esta carrera ingresan cada semestre alrededor de 45 a 50 alumnos, de los cuales terminan en rededor de 20 alumnos llegando a ser siempre un sólo grupo, es difícil formar grupos de matrícula por estudiantes ya que uno de las sistemas que maneja la escuela es tomar materias de otros semestres y cursar las que repruebe para así no retrasarse en la carrera, también por eso se reducen los grupos y las matriculas pierden un orden.

La deserción que ha surgido a lo largo de la carrera ha causado que también las líneas de acentuación no sean cursadas, existen tres líneas las cuales son:

1. Administración de Proyectos Informáticos.
2. Redes y Comunicaciones.
3. Interacción Hombre-Maquina
4. Desarrollo de Software de Aplicación.

3. Desarrollo experimental

Para realizar el estudio aquí mostrado, se aplicó el proceso general de minería de datos, el universo de estudio fueron un total de 497 estudiantes del CUI, en tanto que para el análisis y construcción del modelo de minería de datos, se utilizó el software WEKA (<http://www.cs.waikato.ac.nz/ml/weka/>).

3.1. Adquisición de datos

Con la intención de realizar el estudio aquí mostrado, se desarrolló un instrumento para la recolección de datos (Figura 1) que se aplicó a un total de 497 alumnos de todos los semestres, con reactivos que permitan identificar las posibles causas que pudieran propiciar la deserción de los estudios universitarios.

¿Por qué elegiste la carrera?		¿Con quién vives?	
1.-Me gusta		1.- Papas	
2.-Me Obligaron		2.Parientes/vecinos	
3.-Ganar Dinero		3.-Solo	
¿Quién solventa tus gastos?		¿Trabajas?	
1.-Papa		1.-Sí	
2.-Mama		2.-No	
3.-Ambos			
4.-Yo			
¿Cuál es el tiempo que trabajas?		¿En que trabajas?	
1.-Medio Tiempo		(Pregunta abierta)	
2.-Tiempo completo			
3.-Fines de semana			
¿Qué otra carrera elegirías?		Se te pide que leas cuidadosamente este cuestionario con el fin de realizar una investigación, en las preguntas con diferentes opciones, marcar con una (X), responde lo más honestamente que se te sea posible, por tu participación ¡Gracias!.	
1.-Psicologia			
2.-Arquitectura			
3.-Diseño Grafico			
4.-Derecho			
5.-Administracion			
6.-Contaduria			
7.-Criminologia			
8.-Agronomia			
9.-QFB			
10.-Lenguas			
11.-Comunicacion			
12.- Otra			

Fig. 1. Formato para la aplicación de la encuesta.

3.2. Pre-procesamiento de Datos

En esta etapa se debe realizar una limpieza a los datos, i.e. obtener datos sin valores nulos o anómalos que pudieran obtener patrones de calidad. Los datos obtenidos de las encuestas, fueron analizados para identificar inconsistencia en ellos utilizando el sistema Weka. Este proceso solo se realizó para las preguntas con opción múltiple dándole valor 1 a la respuesta elegida y 0 a las demás opciones. Los atributos

seleccionados en su mayoría no contenían valores nulos ni anómalos (*outliers*), pero en aquellos casos que se presentaban, estos fueron reemplazados utilizando técnicas estadísticas, tales como la media y la moda o derivando sus valores a través de otros, dependiendo del tipo de datos [4].

3.3. Análisis de datos

La Figura 2 muestra el análisis de frecuencias de algunas de las variables estudiadas. Como se puede observar la mayoría de los alumnos que ingresan en la carrera de Ingeniería en computación son hombres con un 69.4% mientras que el 30.6% restante son mujeres. De éstos, la mayor parte son solteros y solo 9 son casados, lo que de algún modo indica que los compromisos familiares no pudieran ser el factor determinante para la deserción.

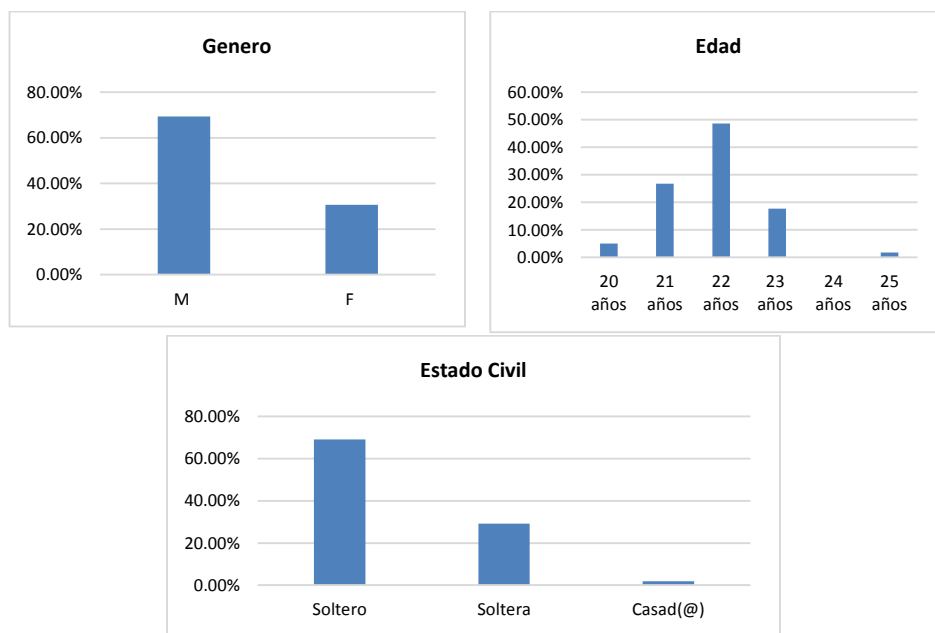


Fig. 2. Análisis de frecuencia de las variables Género, Edad y Estado civil.

Al preguntar el motivo de elección de la carrera, los encuestados manifiestan que en su mayoría (94.4%) la eligieron por agrado personal, en tanto que el porcentaje restante por considerar que sería redituable una vez finalizada. Por otro lado, para identificar el impacto potencial que pudiera tener el aspecto económico en la continuidad de los estudios, la encuesta reveló que la mayoría de los

estudiantes no trabaja (Figura 3b), a los cuales los padres les solventan sus gastos, predominantemente ambos o el padre (Figura 3a). Respecto a los trabajos que tienen los estudiantes, es difícil establecer un predominio de éstos, ya que es muy diverso.

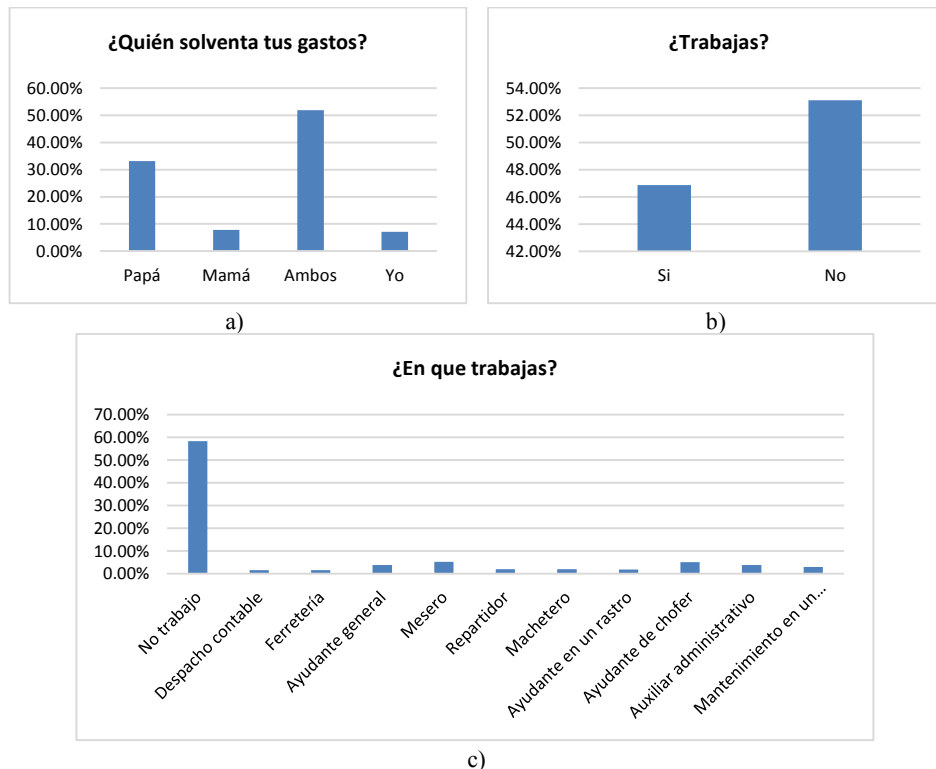


Fig. 3. Fuente de ingresos de los estudiantes encuestados.

3.4. Clasificación

Como se mencionó anteriormente, tanto para el proceso de limpieza, como para la clasificación se utilizó WEKA. Dado que los datos incluidos en el conjunto de datos son en su mayoría categóricos, se optó por utilizar árboles de decisión. De los algoritmos disponibles, se utilizó el programa J48 correspondiente al algoritmo C4.5 [5]. En su ejecución se utilizaron las especificaciones que por default tiene WEKA, así como el método de validación cruzada estratificada.

El procedimiento para generar el árbol consiste en seleccionar un atributo como raíz, y crear una rama con cada uno de los valores posibles de dicho atributo; con cada

rama resultante se realiza el mismo proceso. En cada nodo se debe seleccionar un atributo para seguir dividiendo, y para ello se selecciona aquel que mejor separe los ejemplos de acuerdo a la clase.

En el análisis de la Figura 4 se observa que los alumnos por grupo de edad que entraron a la carrera por decisión propia o porque ven beneficios económicos, de igual modo, se incluyen las formas de solventar sus gastos (por sus padres o de forma personal).

```
Gastos = PAPA: ME GUSTA (165.0)
Gastos = AMBOS: ME GUSTA (258.0/8.0)
Gastos = YO
| Edad <= 22: GANAR DINERO (20.0)
| Edad > 22: ME GUSTA (15.0)
Gastos = MAMA: ME GUSTA (39.0)

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      489          98.3903 %
Incorrectly Classified Instances     8            1.6097 %
Kappa statistic                     0.8251
Mean absolute error                 0.0315
Root mean squared error             0.1255
Relative absolute error              29.1252 %
Root relative squared error          54.4193 %
Coverage of cases (0.95 level)      98.3903 %
Mean rel. region size (0.95 level)  50 %
Total Number of Instances           497

=== Detailed Accuracy By Class ===

      TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
      -----  -
      1         0.286   0.983     1       0.992     0.879    ME GUSTA
      0.714    0         1         0.714   0.833     0.879    GANAR DINERO
Weighted Avg.  0.984    0.27     0.984    0.984   0.983     0.879
```

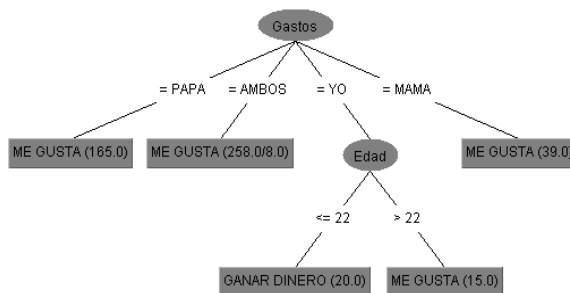


Fig. 4. Estudiantes que les gusta la carrera y quienes solventan sus propios gastos.

En el árbol resultante (Figura 4) se puede observar que, con un 1.6% de error, la predicción indica que los alumnos que solventan sus gastos de la carrera buscan

terminar la carrera para un mejor beneficio económico ya que se limita al tener que trabajar y son cuestiones por las que pueden abandonar sus estudios.

Por otro lado, el árbol de la Figura 5 muestra que los estudiantes que dependen económicamente tan solo de su mamá y su edad está en el rango de 20 a 23 años siguen solteros y los que rebasan los 23 años ya son casados pero siguen dependiendo del apoyo de su madre. Esto último permite indicar que casarse a mitad de la carrera es un factor ya que si se desea continuar con los estudios el apoyo depende de su madre, siendo esto un factor de riesgo para la deserción.

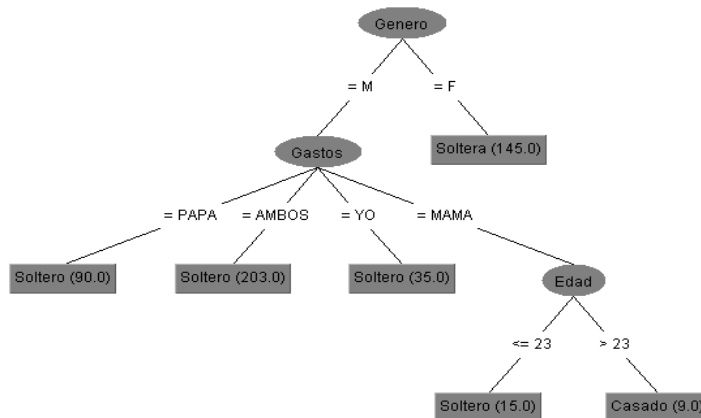
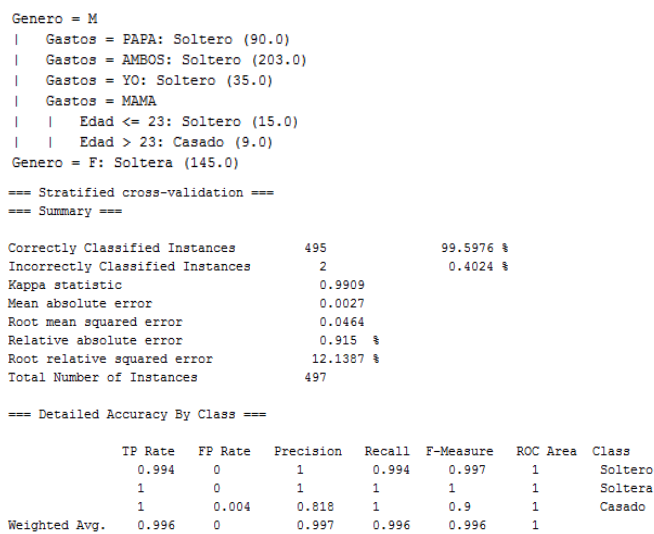


Fig. 5. Clasificación por género, estado civil y forma de solventar sus gastos.

Con la intención de analizar la relación existente entre las variables de “Estado Civil”, “Trabajan” y “Carreras”, el árbol de la Figura 6 muestra que de un total de 20 alumnos entre hombres y mujeres solo 11 escogerían la carrera de Diseño Gráfico, y los 9 restantes son mujeres, si escogieran la carrera de Lenguas la mayoría serían solo las mujeres y trabajan, siendo un total de 15 mujeres y 33 hombres que también estarían en la carrera pero en este caso no trabajarían y siendo el caso de que escogieran alguna otra carrera solo 119 hombres y 24 mujeres trabajarían medio tiempo, dando a entender que estando en otra carrera estarían realizando el trabajo para solventar sus gastos personales y continuar sus estudios.

```

Carreras = COMUNICACION: Soltero (44.0)
Carreras = OTRA
| Trabajan = NO: Soltera (116.0/50.0)
| Trabajan = SI: Soltero (132.0/37.0)
Carreras = DISEÑO GRAFICO: Soltera (20.0/11.0)
Carreras = LENGUAS
| Trabajan = NO: Soltero (33.0)
| Trabajan = SI: Soltera (15.0)
Carreras = QFB: Soltero (87.0)
Carreras = ARQUITECTURA: Soltera (35.0/17.0)
Carreras = CONTADURIA: Soltero (15.0)

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      371          74.6479 %
Incorrectly Classified Instances    126          25.3521 %
Kappa statistic                    0.434
Mean absolute error                 0.1891
Root mean squared error             0.3096
Relative absolute error             64.4932 %
Root relative squared error         80.9931 %
Total Number of Instances          497

=== Detailed Accuracy By Class ===

      TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
      0.816   0.331   0.846     0.816   0.831     0.836   Soltero
      0.607   0.196   0.561     0.607   0.583     0.815   Soltera
      0.333   0.012   0.333     0.333   0.333     0.984   Casado
Weighted Avg.  0.746   0.286   0.753     0.746   0.749     0.833
    
```

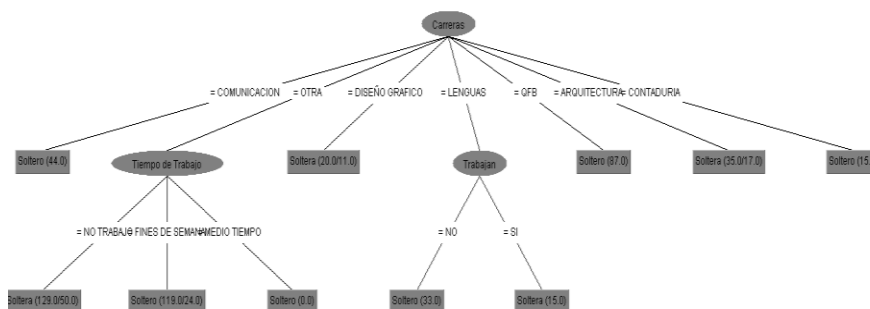


Fig. 6. Clasificación de los alumnos en caso de haber escogido otra carrera viendo en qué carrera solventarían sus gastos ellos mismos.

En el análisis de la Figura 7 se observa que los alumnos que tal vez estudiarían otra carrera serían 145 solteros que mantuvieran algún trabajo, mientras que 50 no trabajan, En el caso de las mujeres 103 mujeres tendrían algún trabajo y 37 no trabajan. Al existir un mínimo de alumnos casados, es posible indicar que el abandono de la carrera no sería especialmente por cuestiones de matrimonio, sino por otras razones que surjan en el transcurso de la carrera.

```

Carreras = COMUNICACION: SI (44.0/11.0)
Carreras = OTRA
| Estado Civil = Soltero: SI (145.0/50.0)
| Estado Civil = Soltera: NO (103.0/37.0)
| Estado Civil = Casado: SI (0.0)
Carreras = DISEÑO GRAFICO: NO (20.0)
Carreras = LENGUAS
| Estado Civil = Soltero: NO (33.0)
| Estado Civil = Soltera: SI (15.0)
| Estado Civil = Casado: NO (0.0)
Carreras = QFB: SI (87.0/34.0)
Carreras = ARQUITECTURA: NO (35.0)
Carreras = CONTADURIA: NO (15.0)

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      365      73.4406 %
Incorrectly Classified Instances    132      26.5594 %
Kappa statistic                    0.4744
Mean absolute error                 0.3461
Root mean squared error             0.4176
Relative absolute error             69.4856 %
Root relative squared error         83.6843 %
Total Number of Instances          497

=== Detailed Accuracy By Class ===

      TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
      0.64     0.159    0.82     0.64     0.719     0.776    NO
      0.841    0.36     0.674    0.841    0.748     0.776    SI
Weighted Avg.  0.734    0.253    0.752    0.734    0.733     0.776
    
```

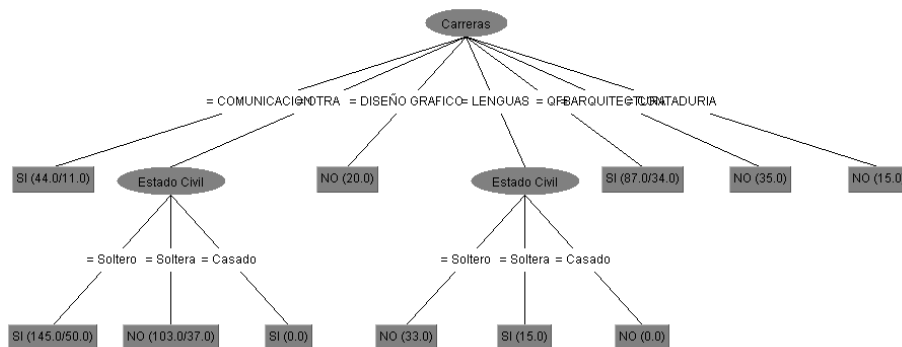


Fig. 7. Casos de haber escogido otra carrera si trabajan o no.

En el árbol de la Figura 8 se analiza la relación existente entre estudiar otra carrera y la forma de solventar sus gastos. Este diagrama muestra cómo se solventan los gastos en caso de darse de baja la carrera. Los resultados muestran con un 2.0% de error que, los gastos seguirían solventados por sus padres en la mayoría de los casos.

```

Carreras = COMUNICACION
| Gastos = PAPA
| | Trabajan = NO: PAPAS (11.0)
| | Trabajan = SI: SOLO (19.0)
| Gastos = AMBOS: PAPAS (14.0)
| Gastos = YO: PAPAS (0.0)
| Gastos = MAMA: PAPAS (0.0)
Carreras = OTRA: PAPAS (248.0)
Carreras = DISEÑO GRAFICO: PAPAS (20.0)
Carreras = LENGUAS: PAPAS (48.0)
Carreras = QFB: PAPAS (87.0/10.0)
Carreras = ARQUITECTURA: PAPAS (35.0)
Carreras = CONTADURIA: PARIENTES (15.0)

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      487          97.9879 %
Incorrectly Classified Instances    10           2.0121 %
Kappa statistic                    0.8638
Mean absolute error                0.0239
Root mean squared error            0.1097
Relative absolute error            21.2651 %
Root relative squared error        46.7918 %
Total Number of Instances          497

=== Detailed Accuracy By Class ===

      TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
      1         0.227   0.978     1       0.989     0.973    PAPAS
      0.655    0         1         0.655   0.792     0.961    SOLO
      1         0         1         1         1         1        PARIENTES
Weighted Avg.  0.98    0.207    0.98     0.98    0.978    0.973
    
```

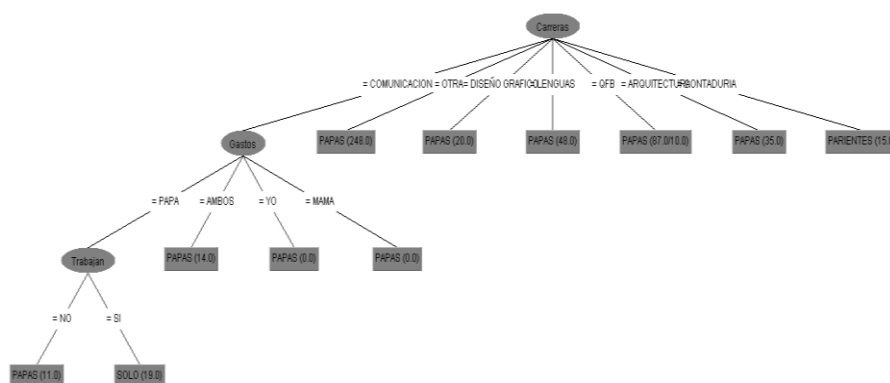


Fig. 8. Clasificación de alumnos que eligieran en caso de haber escogido otra carrera viendo en qué carrera y clasificación de gastos tanto por papá.

4. Conclusiones

Con base a los análisis realizados es posible mencionar que los problemas que afectan la deserción de alumnos en la carrera de Ing. En Computación son los siguientes:

1. Un 46% de los alumnos trabaja en el cual solo un 7% es para solventar sus propios gastos, tomando en cuenta que si los alumnos también tienen que aportar dinero a su familia casi a mediados de semestre abandonarían la carrera por seguir ayudando económicamente a su familia. No obstante, en la mayoría del total de los alumnos los padres son quienes solventan sus gastos, al parecer con independencia del hecho de que trabajen o no.
2. Los alumnos abandonan la carrera ya que también no era lo que ellos esperaban, lo cual no cumple con sus expectativas a futuro y en determinado momento desertan y escojan otra carrera que entre las que más les atraen son arquitectura, Diseño Gráfico, Lenguas, Comunicación y Contaduría.
3. La mayoría de los alumnos que ingresa a la carrera cuenta solo con el apoyo económico de su papá, su mamá o solventa sus gastos el mismo, es la cuestión por la que abandonarían la carrera ya que la carga económica en una persona suele ser muy grande y difícil, tomando en cuenta que se tienen gastos externos como luz, agua, gas entre otros.

Es fundamental se sigan buscando formas en las cuales se apoyen más a los alumnos en cuestiones económicas, esto con la finalidad de que puedan terminar la carrera más alumnos y así se gradúen más ingenieros en Computación.

Agradecimientos. Este trabajo ha sido financiado parcialmente por los proyectos TESJo/CC/001 y SDMAIA-014, del Tecnológico de Estudios Superiores de Jocotitlán, y el 3834/2014/CIA de la UAEM.

Referencias

1. Rodríguez-Lagunas, J., Hernández-Vázquez, J. M.: La deserción escolar universitaria en México. La experiencia de la universidad autónoma metropolitana campus Iztapalapa. Actualidades Investigativas en Educación (2008)
2. Ian, H.W., Eibe, F.: Data mining: Practical machine learning tools and techniques. Morgan Kaufmann, San Francisco, CA (2005)
3. Fawcett, T., Provost, F.: Adaptive Fraud Detection. Data Mining and Knowledge Discovery, Vol. 1, No. 3, pp. 291–316 (1997)

José Luis Aguirre Mendiola, Rosa María Valdovinos Rosas, Juan Alberto Antonio Velazquez, et al.

4. Han, J., Kamber, M.: Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers, USA (2006)
5. Quinlan, J. R.: C4.5 Programs for Machine Learning. San Mateo: Morgan Kaufmann. (1993)

Reconocimiento de dígitos escritos a mano mediante métodos de tratamiento de imagen y modelos de clasificación

Luis Miralles Pechuán, Dafne Rosso Pelayo, Jorge Brieva

Facultad de Ingeniería, Universidad Panamericana,
México

{lmiralles,drosso,jbrieva}@up.edu.mx

Resumen. El ROC (Reconocimiento Óptico de Caracteres) es una línea de investigación dentro del procesamiento de imágenes para la que se han desarrollado muchas técnicas y metodologías. Su objetivo principal consiste en identificar un carácter a partir de una imagen digitalizada que se representa como un conjunto de píxeles. En este trabajo realizamos para el ROC un proceso iterativo que consta de cinco fases. Para ello aplicamos varios métodos de tratamiento de imagen, dos métodos de selección de variables y exploramos diversos métodos supervisados de aprendizaje automatizado. Entre los modelos de clasificación destacamos los de Deep Learning por su novedad y su enorme potencial.

Palabras clave: reconocimiento óptico de caracteres (ROC), modelos predictivos y selección de variables, métodos de clasificación, deep learning, métodos supervisados de aprendizaje automatizado.

1. Introducción

El ROC (Reconocimiento Óptico de Caracteres) consiste en identificar un símbolo, que suele ser un número o una letra, a partir de la digitalización de una imagen [1], [2]. Actualmente existen múltiples procesos en los que se aplica el reconocimiento de caracteres.

Algunos ejemplos de ROC pueden ser: automatizar la redirección de cartas en el correo postal, el reconocimiento de las matrículas de los coches en los radares o la digitalización de los apuntes tomados en una clase mediante la escritura con lápiz óptico en una tablet.

Nuestra investigación consiste en efectuar la identificación de dígitos a partir de un conjunto de píxeles que representan números escritos a mano [3]. La aportación que realizamos es una metodología compuesta de 5 fases para la identificación de dichos números. Para ello, aplicamos la reducción de dimensionalidad, la extracción de características de la imagen original, la selección de variables y la evaluación de varios modelos supervisados de aprendizaje automatizado.

Para la selección de variables hemos incorporado técnicas de PCA (Análisis de componentes principales) [4] y RFE (Eliminación Recursiva de Características) [5] que permiten crear modelos más eficientes para el ROC. Con estas técnicas el tamaño del conjunto de datos decrece y el tiempo de clasificación de los caracteres se reduce. En nuestro caso existen algunas entradas que mantienen siempre el mismo valor o que presentan muy poca variación en sus datos. Estas variables serán descartadas.

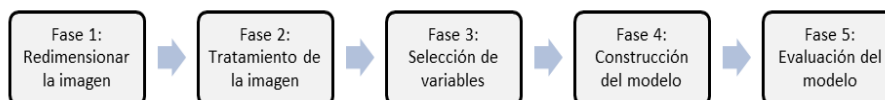


Fig. 1. Fases de la investigación.

El método conformado por 5 fases (Figura 1) incorpora el tratamiento de imágenes así como las fases iterativas correspondientes a la creación de modelos de aprendizaje automatizado.

Para construir un modelo adecuado es importante que las muestras de datos sean de calidad y tengan el suficiente número de muestras. Para la investigación emplearemos el nivel de *accuracy* o precisión. Este valor representa el porcentaje de aciertos en el rango [0,1] para evaluar la calidad de los modelos presentados. Para finalizar, presentaremos una tabla resumen con los resultados de la evaluación.

2. Descripción de la base de datos

Normalmente, el proceso de ROC inicia con la digitalización de las imágenes de escritura realizada a mano que contienen los dígitos, en nuestra investigación partimos de un data set público [3].

Este data set está compuesto por un total de 70,000 muestras [6] de las cuales elegimos 42,000 muestras de manera aleatoria. Cada muestra tiene 785 campos. Un campo que consideramos la salida del modelo representa el dígito y el resto de los campos son las entradas. Las entradas son los píxeles de la imagen que representa el dígito escrito a mano. Cada imagen tiene una resolución de 28x28, lo que hace un total de 784 píxeles. Cada pixel se representa con un número entre el intervalo de 0 y 255 que indica el nivel de gris. De forma que el valor 0 representa el color blanco y el valor 255 el color negro.

3. Fase 1: Redimensionar la imagen

Al número de píxeles que tiene una imagen le llamamos resolución. Cuantos más píxeles usemos para representar una imagen mayor será la resolución. Cada imagen viene representada por un conjunto de píxeles en forma de matriz. Si la imagen es en color tendremos tres valores RGB (Red-Green-Blue) para representar a un píxel. En nuestro caso la imagen está en escala de grises por lo que cada pixel está entre 0 y 255.

Como las imágenes son un insumo de los modelos y estos tienen un número de entradas fijo, debemos homogeneizar las imágenes para que presenten el mismo tamaño. En algunos casos deberemos disminuir el tamaño de la imagen con lo que perderemos información y en otros casos agrandaremos la imagen con lo que deberemos de interpolar los píxeles.

A partir de la imagen tratada, se pueden extraer algunas características que pueden resultar de gran utilidad para aumentar el grado de acierto en las predicciones del modelo. Podemos medir algunas características como: la media de los píxeles, el número de píxeles mayores a un umbral o el número de píxeles que forman una región. También podemos aplicar algoritmos para detectar el tipo de textura o el número de agujeros que tiene cierto símbolo.

En nuestro conjunto de datos, cada imagen está representada por un arreglo bidimensional de píxeles en escala de grises de 28 x 28.

Por lo tanto cada imagen está representada por 785 atributos:

- Dígito en la imagen, su valor es entero desde 0 hasta 9.
- 784 valores en los píxeles (resultado de los 28x28 píxeles), toman valores enteros de 0 hasta 255, siendo el valor 255 la intensidad más oscura (negro) y el 0 la intensidad más clara (blanco).

Trabajar con un conjunto de datos del tamaño 42,000 (número de imágenes) por 197 (número de atributos) es complicado porque cuanto más grande es la dimensión de los datos la creación de los modelos de aprendizaje es más compleja y más lenta. Por ello, es conveniente reducir la dimensión de los datos.

El primer paso para reducir la dimensión de los datos es reducir el tamaño de las imágenes. Esto es, en lugar de que cada imagen sea de 28 x 28 píxeles, trabajaremos con una imagen de 14 x 14 píxeles. Para lograr esto, los valores de cuatro píxeles continuos en la imagen original se promedian para obtener un píxel en la imagen de menor escala (Figura 2).

125	136	252	253
109	95	126	52
125	236	123	145
59	87	38	74

116	171
127	95

Fig 2. Reducción del tamaño de la imagen

Después de realizar este proceso, cada imagen está representada por 197 atributos:

- Dígito en la imagen, su valor es entero desde 0 hasta 9.
- 196 valores en los píxeles (resultado de los 14x14 píxeles), valores enteros de 0 hasta 255.

4. Fase 2: Tratamiento de la imagen

El tratamiento de imágenes es un conjunto de técnicas que se utiliza para mejorar la calidad de la imagen o para facilitar la búsqueda de información en ellas. Algunos ejemplos podrían ser eliminar el ruido, localizar bordes o suavizar la imagen.

Las técnicas más utilizadas y que podemos aplicar a nuestro problema y que aplicaremos en la fase dos son:

- **Binarización:** Consiste en convertir todos los píxeles de una imagen originalmente tengan varios valores a solamente dos tonos: blanco y negro. Para ello se establece un umbral por encima del cual convertiremos en negro y por debajo del cual será blanco. Esta operación es muy frecuente puesto que algunos algoritmos trabajan en esta escala. Las imágenes ocupan un tamaño en memoria muy pequeño puesto que cada píxel se puede representar con un píxel.
- **Fragmentación o segmentación:** Esta técnica consiste en seleccionar una parte de la imagen original. Para ello se desarrollan algoritmos que se basan en la detección de bordes que son pequeños cambios en la intensidad de los colores y otras propiedades. Existen muchos métodos genéricos de fragmentación aunque para cada tipo de imagen se suele aplicar un algoritmo más específico. Por ejemplo, para detectar un cáncer de mama primeramente se aplica un filtro sobre las regiones sospechosas y posteriormente se aplica un algoritmo que evalúa ese conjunto de píxeles.
- **Adelgazamiento de componentes:** Uno de las técnicas más comunes consiste en borrar de manera iterativa los puntos de los contornos. Esta técnica se debe aplicar con cautela para que las imágenes no pierdan su forma original y de manera iterativa. Este método simplifica la identificación de caracteres y permite la extracción de características como la altura de la imagen según los píxeles o el ancho de los píxeles que forman la imagen.
- **Calcular promedio de píxeles:** Este método de tratamiento de imágenes consiste en tomar todas las muestras de un determinado dígito y crear una nueva imagen. Donde cada píxel de esa imagen sea el promedio de todas las imágenes de que representen ese dígito. Por ejemplo, el píxel [1,1] de la imagen promedio de ocho, es el promedio del píxel [1,1] de todas las muestras del número ocho.

El algoritmo procesa las imágenes de todos los dígitos y como resultado obtenemos 10 imágenes con el promedio para todos los píxeles. Esas diez imágenes nos servirán como entradas para crear un nuevo modelo. Pero su tasa de acierto es muy baja por lo que deberemos encontrar un método mejor.

- **Eliminación de filas y columnas en blanco:** Uno de los procedimientos que vamos a realizar en el tratamiento de imágenes es eliminar aquellas filas y columnas que estén blancas y casi blancas. Es decir vamos a eliminar los contornos de todas las imágenes tanto del data set ya que aportan poca información.

Para realizar esto aplicamos un método que suma los píxeles, como el blanco se representa con cero si la suma no es mayor que un determinado umbral eliminaremos esa fila. Lo mismo para las columnas.

- **Selección de Características del objeto:** En esta etapa se extraen características del objeto presente en la imagen después de una binarización simple (Se tomaron los píxeles mayores que cero). El objeto en la imagen es caracterizado mediante un vector con las siguientes componentes: Área, Centroide (componente x,y), Longitud del eje mayor, Longitud del eje menor, Excentricidad, Número de Euler, Diámetro equivalente, Solidez,

Perímetro, Centroide proporcional al nivel de gris (Componentes x, y), Grado, Intensidad Máxima, Intensidad Mínima, e Intensidad promedio. Este vector será la entrada al clasificador en los siguientes procesos.

5. Fase 3: Selección de variables

Una vez completadas las fases anteriores utilizaremos PCA y RFE para seleccionar las características más importantes y descartar las menos relevantes.

El PCA es una técnica estadística empleada en el descubrimiento de patrones en conjuntos de datos con una alta dimensionalidad. Consiste en la selección de componentes que otorgan una mayor aportación de información para el modelo.

- **Información mutua y PCA:** La entropía [4] es una medida de incertidumbre en una variable aleatoria discreta, esto es, mide que tan uniforme es una variable. En este caso, si la variable aleatoria tiene una distribución uniforme, la entropía es máxima. Para calcular la entropía se utiliza la siguiente función:

$$H(X) = - \sum_x P(x) \ln(P(x))$$

La información mutua es una medida de dependencia entre dos variables aleatorias discretas, esto es porque mide la reducción de la incertidumbre (entropía) de una variable, de acuerdo al conocimiento de otra variable. Para el cálculo de la información mutua se utilizan las siguientes ecuaciones:

$$H(X, Y) = - \sum_x \sum_y P(x, y) \ln(P(x, y))$$

$$IM(X, Y) = H(X) - H(Y) - H(X, Y)$$

En el caso de la clasificación de los dígitos manuscritos, las variables más importantes para generar los modelos de clasificación deben coincidir con los píxeles que reducen la incertidumbre de la clase. Esto es, las variables de los píxeles que tengan mayor información mutua. Por lo tanto se evalúa la información mutua de 196 variables y se seleccionan las 150 variables con mayor información mutua. Al analizar este proceso se observa que algunas de las variables descartadas coinciden con los píxeles de los bordes de la imagen, esto es lógico debido a que los bordes generalmente no contienen información importante.

Método recursivo de selección: El método RFE (*Recursive Feature Selection*) o eliminación recursiva de características es un método orientado a la selección de variables. Este método es de tipo *wrapper*, es decir, se van construyendo modelos a partir de una combinación de entradas. [5].

En primer lugar el algoritmo clasifica las entradas del modelo de mayor a menor relevancia. Para ello, construye modelos con cada una de las variables de forma independiente y evalúa los resultados.

Posteriormente va construyendo modelos de la siguiente manera: En primera instancia con la primera variable, en segunda instancia con la primera y segunda variable, en tercera instancia con la primera, la segunda y la tercera y así

sucesivamente para todas las variables. La salida del algoritmo es una gráfica con todas las agrupaciones pero señalando la mejor combinación.

A partir de los resultados obtenidos en la evaluación de modelos se van descartando aquellas variables que son menos importantes o que generan ruido. El objetivo es doble: quedarnos con un número de variables más reducido y elegir la combinación que nos dé mejores resultado. Esto nos permitirá construir modelos más sencillos y más precisos. La aplicación de RFE redujo el número de variables de 196 a 148.

6. Fase 4: Construcción del modelo

6.1. Metodología en la construcción de los modelos

Para la construcción y evaluación de modelos utilizaremos una librería llamada Caret. Caret responde a las siglas: “*Classification And REgression Training*” [7], [8]. Esta librería permite construir modelos en pocas líneas de código y de forma muy sencilla.

A continuación mostramos el código necesario para la construcción de un modelo empleando *Random Forest*:

```
1) control = trainControl (method = "cv" , number = 10, classProbs = T)
2) grid = expand.grid ( .mtry = c ( 5, 10, 25, 50, 75, 100 ) )
3) modeloRForest = train ( Entradas, Salida, method = "rf", tuneGrid =
grid ,ntree = 1000 ,trControl = control )
```

En la primera instrucción `trainControl` sirve para indicar el número de modelos que ejecutaremos por cada configuración. En la segunda instrucción “`expand_grid`” indica el número de árboles de tipo Random Forest que se van a utilizar. La tercera instrucción sirve para crear y evaluar el modelo. Esto lo hacemos con la función “`train`” que tiene como parámetros: una matriz con las entradas, un arreglo con las salidas y el método que se usará en este caso el Random Forest. El número de árboles que emplearemos serán 1000. Las variables “`control`” y “`grid`” las hemos explicado en las dos primeras instrucciones.

A continuación presentamos el algoritmo genérico para la construcción de los modelos:

Algoritmo de construcción de métodos

- Leer el Data Set
 - Seleccionar variables más significativas mediante Random forest o PCA
 - Desde modelo: 1 hasta N
 - Crear conjuntos de muestras diferentes mediante CV
 - Desde configuración 1 hasta configuración N
 - Crear varios modelos con una parte Training Set
 - Probar el modelo con la partición restante
 - Fin Desde
 - Elegimos la media de los modelos y el mejor resultado
 - Fin Desde
 - Clasificar los métodos según los mejores resultados
- Fin del Algoritmo

7. Métodos de clasificación utilizados

Para la construcción de modelos seleccionamos algunos de los clasificadores más conocidos como son: Random Forest, SVM, KNN, Redes Neuronales y experimentamos con métodos de aprendizaje profundo Deep Learning.

Deep Learning: Es un grupo de algoritmos que implementan una metodología basada en crear modelos abstractos a partir de las entradas [9]. Estos algoritmos utilizan una serie de técnicas de aprendizaje orientadas a extraer características de los datos de manera jerárquica. Es decir, las características de nivel superior se forman a partir de las características de los niveles inferiores. Esta técnica permite construir modelos predictivos más precisos y con mejores resultados para un amplio número de problemas [10].

Por ejemplo, una imagen en Deep Learning se puede representar como un conjunto de bordes, una serie de formas o un número de huecos. En lugar de simplemente una combinación de píxeles como hacen la mayoría de los algoritmos.

Estas técnicas de aprendizaje aplican transformaciones no lineales para hacer representaciones de los datos. Para ello utilizan un conjunto de capas concatenadas donde la salida de una capa es la entrada de la siguiente capa. Cuanto más grandes sean las capas y mayor sea su número más aumenta la capacidad de abstracción.

Algunos algoritmos de Deep Learning están basados en la neurociencia y simulan el comportamiento de las neuronas en el cerebro y en el sistema nervioso. El algoritmo trata de establecer una relación entre un estímulo y sus respuestas.

A diferencia de los algoritmos tradicionales que se basan en desarrollar métodos muy específicos para resolver problemas concretos. El Deep Learning trata de crear técnicas útiles orientadas a problemas genéricos.

Las aplicaciones de estos algoritmos pueden ir desde la detección de fraude en las tarjetas de crédito a la predicción de resultados bursátiles. Estos algoritmos dan buenos resultados en el reconocimiento de imágenes y en el procesamiento del lenguaje natural.

Deep Learning H2O: El H2O Deep Learning es un paquete de R Studio que se basa en redes neuronales multicapa con alimentación hacia adelante. Estas redes se entrenan con descenso de gradiente estocástico utilizando retropropagación.

Este modelo descarga a través de Internet una serie de datos empleados en la configuración de los nodos. Estos datos son el resultado de la aplicación del algoritmo a muchos problemas a nivel mundial. Es posible descargar bajo pago ciertas librerías que mejoran sensiblemente tanto los resultados como la rapidez en la ejecución de estos algoritmos.

Random forest: Son una combinación de árboles predictivos donde para cada árbol k se genera un vector aleatorio γ_k , creado de manera independiente a los $\gamma_1 \dots \gamma_{k-1}$ vectores y con la misma distribución empelada en cada árbol k del conjunto (*forest*). El árbol se construye usando el conjunto de entrenamiento y γ_k , dando como resultado un clasificador $h(x, \gamma_k)$ donde x es el vector de entrada. Se selecciona la clase más popular para x , después de que fueron generados un numero suficientemente grande de arboles. [11]

KNN (K-Nearest neighbors): Es un algoritmo de clasificación basado en el cálculo de la distancia más cercana, normalmente distancia Euclidiana, entre los

puntos del conjunto de datos a un punto dado, donde K es el número de vecinos cercanos seleccionados. [12]

SVN (Support Vector Machines): es un clasificador que separa los datos por medio de hiperplanos. En él se busca por el hiperplano óptimo, el cual, es el hiperplano que brinda la máxima distancia desde los patrones de entrenamiento más cercanos, los SV son entonces los vectores más cercanos a estos patrones [13].

8. Fase 5: Evaluación del modelo y muestra de resultados

La evaluación consiste en medir el grado de precisión de cada modelo generado. Esta fase informa si la dimensionalidad elegida es adecuada para la construcción del modelo. Para medir la calidad de los modelos utilizamos el *accuracy* de cada método. Para calcular la exactitud sumamos los elementos de la matriz que forman parte de la diagonal principal y los dividiremos entre el total de elementos.

Podemos apreciar en la Tabla 1, los resultados de aplicar el método SVM con 10.000 muestras de entrenamiento y 15.000 para el test.

Tabla 1. Matriz de confusión con el método SVM

		Valores predichos									
		0	1	2	3	4	5	6	7	8	9
Valores Reales	0	1425	0	6	2	2	4	7	1	5	5
	1	0	1677	3	2	3	2	0	6	6	1
	2	1	7	1421	17	14	3	0	11	5	2
	3	2	0	9	1503	2	29	0	7	11	8
	4	2	2	3	1	1413	6	3	8	9	25
	5	7	0	0	18	2	1279	7	0	10	5
	6	4	3	2	2	9	12	1454	0	6	0
	7	0	6	15	9	7	1	0	1507	3	18
	8	6	1	11	29	2	10	3	3	1385	14
	9	1	2	4	5	28	3	0	19	10	1398

Tabla 2 Fases en el procesamiento de imágenes

Fase 1: Reducción imagen	Fase 2: Tratamiento de imagen	Fase 3: Selección de variables	Fase 4: Construcción de modelos	Fase 5: Resultados
No hacer nada	Ancho de píxeles, Altura píxeles	No extraer características	KNN	0,9425
Imagen 14x14	Quitar filas en blanco	PCA	Random forest	0,9315
Imagen 28x28	Binarizar	RFE	Neuronal Network	0,9625

La exactitud la podemos calcular con la siguiente fórmula:

$$Accuracy = \frac{\sum_{i=1}^K F_{ii}}{N}$$

Tabla 3. Comparativo de los resultados de los modelos para imagen de 28x28

Reducción Imagen	Tratamiento imagen	Selección de Variables	Modelo Predictivo	Parámetros modelo	Resultado
Imagen 28x28	Aplicación de múltiples métodos como densidad, anchura, tamaño, ...	RFE	Random Forest	Árboles: 1000 - mtry: 75	0.7373
				Árboles: 5	0.171
	Promedio de píxeles	No hacer selección	Random Forest	Árboles: 20	0.1615
				Árboles: 50	0.3011

Tabla 4. Comparativo de los resultados de los modelos para imagen de 14x14

Reducción Imagen	Tratamiento imagen	Selección Variables	Modelo Predictivo	Parámetros modelo	Resultado	
Imagen 14x14	Sin tratamiento	PCA	Random Forest	Árboles: 1000 - mtry: 75	0.9284	
				Árboles: 2000 - mtry: 75	0.9650	
		RFE	Random Forest	Árboles: 1000 - mtry: 2	0.9541	
				Árboles: 1000 - mtry: 75	0.9653	
				Árboles: 1000 - mtry: 148	0.9516	
				SVM con núcleo polinómico	degree = 3, scale = 0.1 and C = 0.25.	0.9722
		Neural Networks con extracción de características	size = 5 and decay = 0.1	0.7518		
	Reducción de bordes	Todas	Random Forest	Árboles: 1000 - mtry: 75	0.9330	
				Deep Learning	TanhWithDropout	0.4363
					Tanh	1
			RectifierWithDropout	0.9998		

En la matriz de confusión cada columna representa el número de predicciones de cada clase y cada fila representa las instancias de la clase real. La precisión o *accuracy* del modelo es: 0.9641.

Los resultados obtenidos en el segundo problema los representamos en la tabla 2. El experimento fue realizado utilizando un total de 42,000 muestras con un CV de 10 particiones. Los resultados se muestran en tabla 2.

En la tabla 3 mostramos el nombre del modelo, los parámetros que lo configuran y el promedio de la precisión de los 10 modelos creados.

En la tabla 4 mostramos los resultados de algunos métodos tradicionales y de los métodos Deep Learning. Estos métodos son los que mejor se adaptan a estas imágenes. Posiblemente por su capacidad de abstracción.

En la tabla se muestra en cada columna los métodos aplicados en cada una de las fases. Como se aprecia en los resultados los métodos Deep Learning son los que tienen mayor precisión.

9. Conclusiones

Después de haber trabajado con R Studio y la librería Caret lo calificamos como un entorno muy adecuado para resolver problemas relacionados con la construcción y evaluación de modelos predictivos.

Las principales ventajas de esta herramienta son que permite crear modelos con distintas configuraciones, la selección automática de la mejor combinación y la muestra de resultados en distintas métricas. Además, contempla la opción de *cross-validation*, que crea diferentes particiones para crear varios modelos y así hacer cálculos más robustos.

El hecho de que sea software gratuito representa una gran ventaja porque permite a estudiantes y a personas de la comunidad científica disfrutar de una herramienta con una amplia gama de funciones. Un inconveniente de R es que no tienen ningún elemento que indique la estimación del tiempo restante. Esto dificulta hacer una planificación eficiente para probar los modelos. Tampoco da garantías en caso de que algún error ocurra como indica claramente al inicializar el intérprete de comandos.

La documentación de R no recoge todos los detalles de las funciones por lo que en algunos casos resulta laborioso encontrar la configuración de algunos parámetros. Esto nos ha sucedido en el caso de las funciones RFE. Por último, vemos que es posible implementar esta herramienta con otros lenguajes como Java.

Consideramos que el redimensionamiento de la matriz para trabajar con imágenes más reducidas ha permitido ahorro de tiempo y complejidad en la creación de los modelos.

Podemos comprobar en la tabla 4, que nuestro método propuesto basado en la reducción de dimensionalidad y la selección de características permitieron obtener un acierto del 100% en el reconocimiento de los dígitos escritos a mano. Aplicando *Random Forest* en el conjunto original sin reducción de la imagen se obtiene una exactitud de 0.7373 mientras que aplicando los elementos descritos en las fases del método se logra mejorar a una precisión de 0.9653, mientras que los algoritmos de Deep Learning lograron alcanzar un 0.9998 y un 1 de exactitud. Por lo que concluimos que este método tiene un gran futuro en los modelos predictivos.

10. Trabajos futuros

El sistema que hemos creado parte de una fase en la que se localizan los dígitos y se centran en una imagen. De forma que partimos de imágenes ya recortadas que facilitan en gran medida el reconocimiento. Pero en la vida real los caracteres no están aislados sino que se encuentran dentro de textos. Una manera de segmentar este problema es descomponer los textos en pequeñas imágenes que reconozcan solamente un dígito.

Un siguiente paso en la investigación sería aplicar nuestro modelo a imágenes en color. Una forma sencilla sería pasar las imágenes a escala de grises y aplicar nuestra metodología para evaluar los resultados.

De la misma forma que hemos conseguido crear un modelo con una gran precisión para el reconocimiento de números se podría evaluar la misma metodología para el reconocimiento de letras. El estudio sería interesante, ya que si hemos alcanzado un nivel de precisión alto en el reconocimiento de dígitos esperamos buenos resultados para el reconocimiento de letras.

Una línea de investigación interesante para tener en cuenta la estabilidad de los algoritmos sería incrementar el ruido en las imágenes puesto que sabemos que en la vida real las imágenes están sujetas a todo tipo de distorsión. Entendemos por esto que el papel tenga otro color de fondo, que los números puedan ser diferentes, etc. Para ello se aplicaría un algoritmo de generación de ruido sobre las imágenes del conjunto de datos y se volvería a crear un modelo para evaluarlo.

Teniendo en cuenta los resultados obtenidos con los métodos de Deep Learning nos parece interesante seguir profundizando en este tipo de algoritmos. Pensamos que puede ser de interés general un estudio comparativo de los métodos Deep Learning. Para ello, utilizaremos varios conjuntos de datos públicos y compararemos tanto la precisión de los resultados y como el tiempo necesario para la construcción de modelos.

Bibliografía

1. Mori, S., Nishida, H., Yamada, H.: Optical character recognition. John Wiley & Sons, Inc (1999)
2. Vithlani, P., Kumbharana, C. K.: A Study of Optical Character Patterns identified by the different OCR Algorithms. Department of computer science, Saurashtra University, Rajkot, India (2012)
3. Mohiuddin, K., Mao, J.: A comparative study of different classifiers for handprinted character recognition. Pattern Recognition in Practice, Vol. IV, pp. 437–448 (2014)
4. Bro, R., Smilde, A. K.: Principal component analysis. Analytical Methods, Vol. 6, no. 9, pp. 2812–2831 (2014)
5. Zhang, X., Lu, X., Shi, Q., Xu, X. Q., Hon-chiu, E. L., Harris, L. N., Wong, W. H.: Recursive SVM feature selection and sample classification for mass-spectrometry and microarray data. BMC bioinformatics, Vol. 7, no. 1, p. 197 (2006)
6. Le Cun, Y., Cortes, C.: MNIST handwritten digit database. AT&T Labs. En línea: <http://yann.lecun.com/exdb/mnist> [Último acceso: 15 Enero 2015]
7. Kuhn, M., Wing, J., Weston, S., Williams, A., Keefer, C., Engelhardt, A.: Caret: classification and regression training. R package, Vol. v515 (2012)

8. Kuhn, M.: Building predictive models in R using the caret package. *Journal of Statistical Software*, Vol. 28, no. 5, pp. 1–26 (2008)
9. Schmidhuber, J.: Deep learning in neural networks: An overview. *Neural Networks*, Vol. 61, pp. 85–117 (2015)
10. Weigel, V. B.: *Deep Learning for a Digital Age: Technology's Untapped Potential to Enrich Higher Education*. Jossey-Bass (2002)
11. Breiman, L.: Random Forests. In: *Machine Learning*. Kluwer Academic Publishers, no. 45, pp. 5–32 (2001)
12. Yu, X., Pu, K. Q., Koudas, N.: Monitoring k-Nearest Neighbor Queries Over Moving Objects. In: *Proceedings of the 21st International Conference on Data Engineering* (2005)
13. Duda, R. O., Hart, P. E., Stork, D. G.: *Pattern Classification, USA*: John Wiley & Sons (2012)

Aproximación de funciones con EPWavenets

Alejandro Romero Herrera, Oscar Herrera Alcántara, Victor Manuel Landassuri Moreno

Universidad Autónoma del Estado de México,
Estado de México, México

aromero688@alumno.uaemex.mx
{oherreraa, vmlandassurim}@uaemex.mx

Resumen. En este trabajo presentamos las redes neuronales *EPWavenets* como una arquitectura de red neuronal de funciones de base radial modificada para incluir funciones wavelets que se obtienen con parametrizaciones de filtros de reconstrucción perfecta. Los parámetros son optimizados con un algoritmo evolutivo a fin de mejorar la aproximación de funciones $f(x)$. Los experimentos comparan el desempeño de las *EPWavenets* con redes neuronales que usan la función gaussiana y la función sigmoïdal. Se concluye que las *EPWavenets* logran un alto grado de adaptabilidad y un desempeño globalmente competitivo respecto a las demás redes neuronales.

Palabras clave: Wavelets, redes neuronales, funciones de base radial, algoritmos evolutivos.

1. Introducción

Una de las motivaciones que dieron origen a las redes neuronales artificiales (RNA) fue la inspiración en el modelo conexionista del cerebro humano [11]. Posteriormente, ante los cuestionamientos del cómo y por qué funcionaban las redes neuronales y en vista a hacer un mejor uso de ellas, se estableció la conexión con otras áreas del conocimiento, en particular con el Análisis Funcional que es una rama del Análisis Matemático orientada a estudiar las propiedades de funciones que generan espacios vectoriales. El Teorema de Aproximación Universal (TAU), por ejemplo, permite abordar desde esta perspectiva la capacidad de aproximar funciones a partir de funciones no constantes, acotadas, monótonas crecientes y continuas [1]. El TAU guarda una estrecha relación con las redes neuronales de perceptrones.

Otro tipo de redes neuronales son las de funciones de base radial (RNFBR) cuya expresión como aproximadores de funciones está dada por:

$$f(x) \approx \sum_{i=1}^N w_i \varphi(\|x - x_i\|) + w_0 \quad (1)$$

en donde $x \in R^n$ es el conjunto de datos de entrada, N es el número de unidades de procesamiento o neuronas, $x_i \in R^n$ son los centroides de las funciones de base

radial, w_i son los pesos sinápticos, w_0 es el umbral de desplazamiento, $\|x - x_i\|$ es la métrica de distancia, y $\varphi(\cdot)$ es la función de base radial.

Se han propuesto varias funciones de base radial entre las que se encuentran las funciones: gaussiana, multicuadrática, multicuadrática inversa, y *splines* [4].

Otras funciones que también se han usado como funciones base son las *wavelets*, que dan lugar a las redes neuronales *wavenets* [14][3].

Para aproximar una función $f(x)$ mediante funciones wavelets $\psi(x)$ se puede usar la expresión:

$$f(x) \approx \sum_{i=1}^N w_i \psi_{a_i, b_i}(x) + w_0 = \sum_{i=1}^N w_i \psi\left(\frac{x - b_i}{a_i}\right) + w_0 \quad (2)$$

en donde los $a_i \neq 0$ son parámetros de escalamiento, y los $b_i \in R$ son parámetros de desplazamiento de la función wavelet principal $\psi(x)$.

El problema de entrenamiento de la RNFBR descrita en la ec. (1) radica en determinar el conjunto óptimo de parámetros libres que incluyen: el conjunto de valores a_i , b_i , los pesos sinápticos w_0 y w_i con $i = 1, 2, \dots, N$, donde N es el número de neuronas.

En éste artículo presentamos las redes neuronales *EPWavenets* que usan bases wavelets adaptables obtenidas a partir de filtros de reconstrucción perfecta. En las *EPWavenets* el ajuste de parámetros libres se realiza aplicando un algoritmo evolutivo que minimiza el error al aproximar una función objetivo $f(x)$.

El resto del artículo está organizado de la siguiente manera: En la Sección 2 revisamos conceptos de wavelets, en la Sección 3 se comenta cómo generar múltiples funciones wavelets a partir de filtros paramétricos de reconstrucción perfecta. En la Sección 4 comentamos la relación entre wavelets y redes neuronales. En la Sección 5 describimos las *EPWavenets* y cómo aplicar un algoritmo genético para optimizar sus parámetros libres para aproximar funciones. En la Sección 6 presentamos resultados experimentales y, finalmente, en la Sección 7 presentamos las conclusiones obtenidas además de algunas ideas para continuar en trabajos futuros.

2. Wavelets

Las wavelets son familias de funciones $\psi_{a,b}(x) = a^{-\frac{1}{2}} \psi\left(\frac{x-b}{a}\right)$ con $a \neq 0$, que surgen de una función principal $\psi(x)$ que cumple las condiciones $\int_{-\infty}^{\infty} \psi(x) dx = 0$ y $\int_{-\infty}^{\infty} |\psi(x)|^2 dx = 1$. Las funciones wavelets además cumplen la condición de admisibilidad dada por $\int_{-\infty}^{\infty} \frac{|\hat{\psi}(w)|^2}{|w|} dw < \infty$ lo cual les da la propiedad de localización, que significa que su transformada de Fourier $\hat{\psi}(w)$ se desvanece rápidamente. También se pueden generar wavelets con soporte compacto cuyo valor es cero fuera del intervalo de soporte en el dominio temporal. Una forma de generar funciones wavelets es a través del análisis multiresolución (AMR) [8] en donde se hace uso de funciones de escalamiento $\phi(x)$ ortogonales a las funciones

wavelets $\psi(x)$ y que en conjunto conforman una base del espacio vectorial $L^2(\mathbb{R})$, que es el espacio de todas las funciones f para las cuales $\int_{-\infty}^{\infty} |f(x)|^2 dx < \infty$.

3. Filtros ortogonales de reconstrucción perfecta

De acuerdo con el AMR es posible generar wavelets ortogonales con soporte compacto a través de la parametrización de los coeficientes de un banco de filtros ortogonales h y g de reconstrucción perfecta (FORP). Los FORP constan de un filtro pasabajas h y un filtro pasaaltas g . Un filtro $h = [h_0, h_1, h_2, h_3]$ de longitud $L = 4$, por ejemplo, genera wavelets con soporte compacto en el intervalo $[0, 1]$, y sus coeficientes dependen de un parámetro α de la siguiente manera [10]:

$$\begin{aligned} h_0 &= \frac{1}{4} + \frac{1}{2\sqrt{2}}\cos\alpha & h_1 &= \frac{1}{4} + \frac{1}{2\sqrt{2}}\sin\alpha \\ h_2 &= \frac{1}{4} - \frac{1}{2\sqrt{2}}\cos\alpha & h_3 &= \frac{1}{4} - \frac{1}{2\sqrt{2}}\sin\alpha \end{aligned} \quad (3)$$

en donde $\alpha \in [0, 2\pi)$. Los correspondientes coeficientes del filtro pasaaltas se calculan a partir de los coeficientes h_i mediante la ecuación $g_i = (-1)^i h_{N-i-1}$. En el caso de filtros de longitud $L = 6$ los coeficientes dependen de dos parámetros α y β de la siguiente manera [10]:

$$\begin{aligned} h_0 &= \frac{1}{8} + \frac{1}{4\sqrt{2}}\cos\alpha + \frac{p}{2}\cos\beta, & h_1 &= \frac{1}{8} + \frac{1}{4\sqrt{2}}\sin\alpha + \frac{p}{2}\sin\beta \\ h_2 &= \frac{1}{4} - \frac{1}{2\sqrt{2}}\cos\alpha, & h_3 &= \frac{1}{4} - \frac{1}{2\sqrt{2}}\sin\alpha \\ h_4 &= \frac{1}{8} + \frac{1}{4\sqrt{2}}\cos\alpha - \frac{p}{2}\cos\beta, & h_5 &= \frac{1}{8} + \frac{1}{4\sqrt{2}}\sin\alpha - \frac{p}{2}\sin\beta \end{aligned} \quad (4)$$

en donde $p = \frac{1}{2}\sqrt{1 + \sin(\alpha + \frac{\pi}{4})}$, $\alpha, \beta \in [0, 2\pi)$.

Se puede demostrar que los FORP deben tener una longitud L par, y que el soporte compacto del wavelet es $[0, \frac{L}{2} - 1]$.

Cabe mencionar que existen pocas funciones wavelet con soporte compacto que tengan una expresión analítica. Uno de los casos en los que sí se tiene la expresión analítica es para la función wavelet Haar descrita como:

$$f(x) = \begin{cases} 1 & \text{si } x \in [0, \frac{1}{2}) \\ -1 & \text{si } x \in [\frac{1}{2}, 1) \\ 0 & \text{caso contrario} \end{cases} \quad (5)$$

En otros trabajos con *wavenets*, a diferencia de las *EPWavenets*, se usan wavelets sin soporte compacto o con una $\psi(x)$ fija. En [9] se usa el wavelet $\psi(x) = -e^{-\frac{x^2}{2}} \cos(5x)$ con parámetros de escalamiento y dilatación fijos, y en [14] y [13] se usa la “Derivada-Gaussiana” $\psi(x) = -xe^{-\frac{1}{2}x^2}$.

En el caso de la *EPWavenets* hemos optado por aproximar las funciones wavelets en forma discreta usando el algoritmo en cascada [7] y filtros ortogonales paramétricos, lo cual nos da una amplia gama de posibilidades. A manera de ejemplo considérese la aproximación del wavelet Haar con las ecuaciones dadas en (3) y $\alpha = \frac{\pi}{4}$. La Figura 1 muestra el wavelet Haar aproximado con $A = 16$ valores discretos, en donde el soporte es $[0, 1]$.

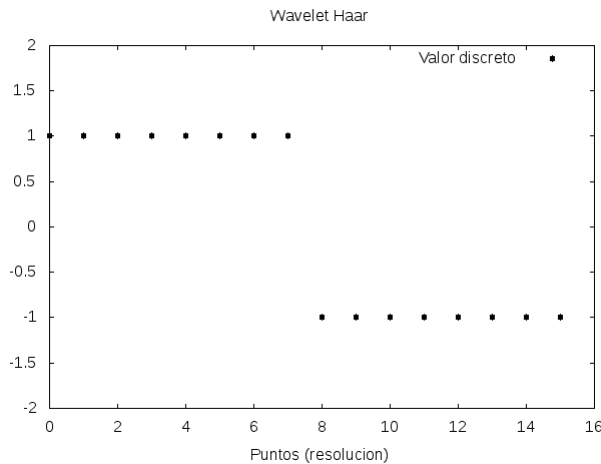


Fig. 1: Aproximación del wavelet de Haar con $\alpha = \frac{\pi}{4}$ y una resolución de $A = 16$ puntos

4. Redes neuronales de funciones de base radial

La arquitectura de las RNFBR define tres capas: la capa de entrada, la capa oculta, y la capa de salida. La capa oculta aplica una transformación no lineal a los datos provenientes de la capa de entrada para llevarlos a otro espacio vectorial que suele tener una mayor dimensión. Al pasar de un espacio a otro de mayor dimensión se explota de mejor manera la capacidad de las redes neuronales en tareas de clasificación y de aproximación de funciones [4]. Esta última tarea es de interés en el presente artículo.

Una RNFBR como aproximador de funciones descompone cada uno de los datos de entrada en varias componentes que pasan por N neuronas, mismas que se suman en la capa de salida. La Figura 2 ilustra la arquitectura de una RNFBR acorde con la ecuación (1), en donde cada muestra de entrada corresponde a un vector $X \in R^n$. Cada neurona incluye una función de base radial (FBR), entre las que se encuentran: la función gaussiana, la función cuadrática, la función cuadrática inversa, la función multicuadrática, y las funciones *splines*. La capa de salida realiza una combinación lineal de las activaciones de las neuronas de la capa oculta considerando los pesos sinápticos w_i . El peso sináptico w_0 corresponde al valor de umbral ($BIAS=1$) que modifica el valor de la salida Y .

De manera análoga al funcionamiento de las RNFBR, las *EPWavenets* utilizan las funciones wavelets como una opción a las FBR. Las funciones wavelets tienen propiedades que no tienen las FBR. Por ejemplo, el soporte compacto es una propiedad relevante toda vez que permite capturar información local de una función, y no interferir con la aproximación global. Las FBR sí permiten identificar información local, sin embargo el número de opciones se restringe a

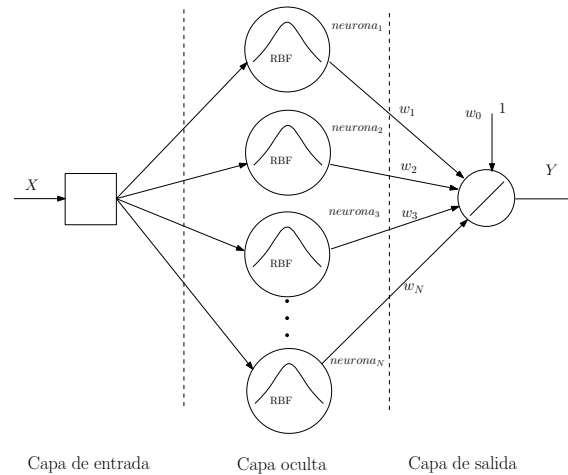


Fig. 2: Una red neuronal de función de base radial

algunas de ellas como son: la función gaussiana, la función cuadrática, la función cuadrática inversa, y los *splines*. Otra propiedad relevante de las wavelets obtenidas con FORP's es la posibilidad de controlar los momentos de desvanecimiento definidos como:

$$\int x^k \psi(x) dx = 0, \text{ para } k = 1, 2, \dots, p \quad (6)$$

y que cuanto mayor sea el valor de p los wavelets son “más suaves”, lo que facilita que funcionen apropiadamente en la aproximación de funciones que tienen varias derivadas. Los wavelets que maximizan los momentos de desvanecimiento para un soporte compacto dado corresponden a las funciones propuestas por Daubechies [2], y que están incluidos dentro de las parametrizaciones de los FORP (ver [5]).

Nótese ciertamente que el uso de wavelets junto con redes neuronales es un tema ya abordado [14,9,13], sin embargo la principal aportación de las *EPWavenets* incluye el uso de wavelets paramétricos optimizados evolutivamente para adaptarlas a un contexto particular y a su vez mantener la adaptabilidad para aproximar diferentes funciones.

5. EPWavenets

La idea fundamental de las redes neuronales *EPWavenets* consiste en incluir funciones wavelets generadas con filtros paramétricos (ver ecuaciones 3 y 4), usar la ecuación de aproximación (2), y aplicar algoritmos evolutivos para determinar el conjunto óptimo de parámetros libres. El uso de wavelets y redes neuronales ya ha sido abordado en otros trabajos. En [9] por ejemplo, se optimizan los pesos w_i y se dejan fijos los parámetros de traslación y escalamiento, lo cual restringe

demasiado la capacidad de aproximación con wavelets. En [14] se propone usar un método de gradiente para optimizar los parámetros w_i junto con los parámetros a_i y b_i . Sin embargo, el problema inherente al uso de métodos de gradiente es que se requiere conocer la función analítica de $\psi(x)$ misma que además debe ser derivable, algo que ciertamente restringe la variedad de funciones wavelets, y conlleva a usar un mayor número de neuronas cuando la función a aproximar no es suave.

También es conocido que determinar el conjunto de parámetros óptimos de los filtros paramétricos para aproximar una función es un problema no convexo [12]. Entonces, dada la complejidad de calcular los parámetros óptimos de los filtros y de los parámetros a_i, b_i, w_i de la red neuronal, se propone el uso de algoritmos evolutivos, por lo que podemos decir que las *EPWavenets* son la combinación de algoritmos evolutivos, wavelets y redes neuronales:

$$EPWavenets = \text{Algoritmos Evolutivos} + \text{Wavelets} + \text{Redes Neuronales}$$

La Figura 3 ilustra una red neuronal *EPWavenet* con los diferentes parámetros involucrados. La función que minimiza el algoritmo evolutivo es el error de

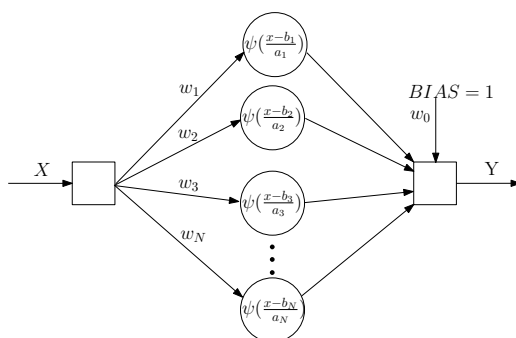


Fig. 3: Red neuronal *EPWavenet*

aproximación entre la salida Y de la *EPWavenet* y los valores de la función objetivo $f(x)$ obtenidos con la entrada X . Tanto X como Y están en la misma dimensión n , así que el error de aproximación (cuadrático medio) puede ser calculado así:

$$Error = \frac{1}{M} \sum_{m=1}^{m=M} \sum_{i=1}^n (x_i(m) - y_i(m))^2 \tag{7}$$

en donde M es el número de muestras de los datos de entrada, $x_i(m)$ es la i -ésima componente del dato de entrada en el instante m , y $y_i(m)$ es la i -ésima componente de la salida de la *EPWavenet* en el instante m . La aproximación con wavelets puede ser suficientemente pequeña de acuerdo con el número de funciones involucradas, y con los valores apropiados de los parámetros de escalamiento

y dilatación. Matemáticamente se expresa así:

$$\left| \left(\sum_{i=1}^N w_i \psi\left(\frac{x - b_i}{a_i}\right) + w_0 \right) - f(x) \right| < \epsilon \quad (8)$$

para $\epsilon > 0$. En particular para un filtro de longitud 4 se tiene un parámetro α , y con N neuronas se generan $3N + 1$ parámetros adicionales que corresponden a los valores de a_i , b_i , w_i , con $i = 1, 2, \dots, N$. Si se usa un algoritmo genético, la codificación de un individuo podría ser así:

$$\alpha | a_1 a_2 \dots a_N | b_1 b_2 \dots b_N | w_0 w_1 w_2 \dots w_N$$

Para un filtro de longitud 6 se tienen dos parámetros α y β , y la codificación en un individuo podría ser así:

$$\alpha \beta | a_1 a_2 \dots a_N | b_1 b_2 \dots b_N | w_0 w_1 w_2 \dots w_N$$

En la siguiente sección ponemos en práctica los conceptos explicados en las secciones anteriores.

6. Experimentos

Con los siguientes experimentos se persigue:

- Explorar la capacidad de aproximación de funciones wavelets generadas con FORP.
- Comparar el desempeño entre las wavelets paramétricas en una arquitectura de *EPWavenets* y funciones gaussianas en una arquitectura de RNFBR al aproximar y predecir funciones.

Aproximación de funciones wavelets paramétricas. En los experimentos se realiza la aproximación de funciones wavelets a partir de parameterizaciones de filtros de longitud $L = 4$ y $L = 6$. Con A muestras de aproximación para la función wavelet $\psi(x)$ con un intervalo para el soporte compacto $[0, \frac{L}{2} - 1]$, y para valores de x dentro del soporte compacto, la mejor aproximación para $\psi(x_r)$ está dada por la r -ésima muestra discreta, donde $r = \lfloor x_r * A / (N - 1) \rfloor$. Obviamente, para valores fuera del soporte $\psi(x) = 0$. Nótese que el valor aproximado del wavelet es devuelto como si $\psi(x)$ fuese una función analítica continua.

Configuración del AG. Existen varios algoritmos evolutivos que pueden usarse con las *EPWavenets*. En los experimentos realizados se optó por usar un algoritmo genético no tradicional [5] caracterizado por:

- Una población con P individuos, en donde P es par, y una población temporal de tamaño $2P$.

- Cruzamiento en parejas entre el i -ésimo y el $(P - 1 - i)$ -ésimo individuos, en donde $i \in [0, \frac{P}{2} - 1]$. Los individuos se ordenan según la medida de aptitud.
- Selección determinista, en donde los P individuos más aptos de la población temporal pasan a la siguiente generación.
- Cruzamiento anular en un solo punto.

Para codificar los parámetros α y β en cada uno de los individuos se utilizó una codificación en binario pesado, con $q = 29$ bits de precisión por parámetro. Para generar valores $V_{\alpha\beta}$ para α y β en un rango de $[0, 2\pi)$ dado el valor en binario pesado $B_{\alpha\beta}$ se usó la ecuación (9):

$$V_{\alpha\beta} = 2\pi \frac{B_{\alpha\beta}}{2^q - 1} \tag{9}$$

Para codificar los valores de w , a y b se usaron $q = 28$ bits de precisión, y un bit de signo. A partir de su valor en binario pesado con signo B_{wab} se obtuvieron valores V_{wab} en el intervalo $(-8192, 8192)$ mediante la ecuación (10):

$$V_{wab} = 8192 \frac{B_{wab}}{2^q - 1} \tag{10}$$

El número de generaciones del algoritmo genético fue $G = 2000$, una población de $P = 50$ individuos, una probabilidad de cruzamiento $P_c = 0.97$ y una probabilidad de mutación $P_m = 0.02$. El tiempo promedio de ejecución del algoritmo genético con los parámetros anteriores para un conjunto de datos de entrada de $M = 1000$ es de 11 minutos, para $M = 5000$ es de 13 minutos y para $M = 9000$ es de 16 minutos, corriendo sobre servidores Sun Fire X2270 con procesadores Xeon de 16 núcleos y 32 GB de RAM con GNU/Linux Debian. En los experimentos se consideraron 13 funciones $f(x)$ cuyos valores se normalizaron (valores en el intervalo $[0, 1]$). La Tabla 1 resume la información de cada función $f(x)$, en donde en la primera columna se muestra el nombre de la función, en la segunda se muestra un identificador $f\#$, y en la tercera columna se indica el número de muestras o puntos para la función $f(x)$ discretizada. El origen de nuestras funciones es el repositorio del software Visual Recurrence Analysis (VRA) [6]. Con las funciones de la Tabla 1 se realizaron los siguientes experimentos:

Tabla 1: Funciones $f(x)$ del repositorio y sus características

Nombre	Brownian Motion	Dow Jones	ECG	Henon	Ikeda	Laser	Logistic	Lorenz	Rosler	Seno	Seno con Ruido	Sun Spots	White Noise
f#	f1	f2	f3	f4	f5	f6	f7	f8	f9	f10	f11	f12	f13
Número de muestras (M)	1000	1000	9000	5000	5000	5000	1000	3500	5000	1000	1000	280	1000

6.1. Experimento 1

Determinar la mejor *EPWavenet* para una función $f\#$, con datos de entrada en una dimensión ($n = 1$), variando el número de neuronas $N = 1, 2, \dots, 10$, y el tipo de filtro (wavelet) con $L = 4$ para un filtro de longitud 4, y $L = 6$ para un filtro de longitud 6, ver ecuaciones (3) y (4). El criterio para elegir la mejor arquitectura fue minimizar el error cuadrático medio (ver ec. (7)), sobre las M muestras. Los resultados se muestran en la Tabla 2 en donde en la primera columna se indica el valor de L y N , y en las demás columnas se indica el tipo de función aproximada identificada por $f\#$. La mejor arquitectura por columna se ha remarcado en negritas para cada caso. El número promedio de neuronas

Tabla 2: Arquitecturas de *EPWavenets* para la aproximación de funciones: $f1$. Brownian Motion, $f2$. Dow Jones, $f3$. ECG, $f4$. Henon, $f5$. Ikeda, $f6$. Laser, $f7$. Logistic, $f8$. Lorenz, $f9$. Rossler, $f10$. Seno, $f11$. Seno con Ruido, $f12$. Sun Spots, $f13$. White Noise

Func	$f1$	$f2$	$f3$	$f4$	$f5$	$f6$	$f7$	$f8$	$f9$	$f10$	$f11$	$f12$	$f13$
L, N	NC	NC	NC	NC	NC	NC	NC	NC	NC	NC	NC	NC	NC
4, 1	1.84E-6	0.00434	0.09305	0.19952	0.17513	2.91E-9	1.21E-5	0.24306	1.18E-5	4.62E-6	1.28E-16	7.23E-4	0.00398
4, 2	0.00345	0.00903	0.07690	0.21114	0.04699	0.03520	0.00183	0.07842	0.02476	0.00285	0.00229	3.98E-5	1.99E-4
4, 3	0.00703	0.00869	0.02102	0.00541	0.06779	0.01058	3.12E-4	0.01952	0.09505	0.00107	1.22E-4	9.59E-7	1.47E-7
4, 4	4.06E-4	0.00594	0.02525	0.02009	0.01489	0.03456	1.65E-4	0.01685	0.01646	0.00204	5.65E-4	1.56E-5	1.17E-4
4, 5	2.87E-4	5.30E-4	0.02084	0.06891	0.00753	0.00669	2.39E-4	0.01551	0.08655	3.58E-5	4.49E-4	9.48E-5	3.63E-4
4, 6	1.27E-5	8.12E-4	0.02941	0.04353	0.01084	0.00444	4.26E-5	0.01015	0.00648	2.47E-4	1.53E-4	7.31E-7	5.97E-6
4, 7	2.28E-5	0.00152	0.03750	0.07567	0.01000	0.01069	2.20E-10	0.01065	0.01655	1.13E-5	3.10E-9	1.16E-5	1.17E-4
4, 8	1.68E-8	0.00649	0.01166	8.23E-5	1.63E-4	0.00436	2.97E-5	8.87E-6	0.02839	1.79E-8	3.64E-4	1.27E-5	5.31E-8
4, 9	1.05E-5	0.00315	0.01743	0.05384	0.05532	0.00418	6.67E-5	0.04234	4.72E-5	2.25E-5	8.31E-5	9.05E-7	2.36E-5
4, 10	3.50E-5	0.00133	0.03197	0.08349	0.01160	0.01280	4.18E-4	0.00818	0.01212	6.54E-4	1.08E-5	9.22E-6	3.08E-5
6, 1	0.00519	0.00587	0.08905	0.12375	0.19189	0.02997	9.47E-4	0.10162	0.06970	9.76E-4	2.11E-5	3.21E-4	0.01368
6, 2	0.02138	0.00246	0.02127	0.11266	0.00378	0.02125	2.22E-4	0.00426	0.03568	0.01482	0.00128	1.31E-4	0.00266
6, 3	0.00258	0.01542	0.03566	0.07052	0.00833	0.02580	2.16E-5	0.03497	0.05618	5.82E-5	0.00743	0.00109	2.02E-4
6, 4	2.69E-5	0.00432	0.03510	0.08667	0.01482	0.00532	1.09E-4	0.01528	0.04018	5.06E-5	1.99E-4	4.19E-4	1.78E-4
6, 5	5.94E-5	6.13E-4	0.01794	0.02906	0.04455	7.26E-4	4.59E-5	6.33E-4	4.11E-4	2.84E-5	1.39E-4	2.10E-4	5.51E-4
6, 6	2.91E-4	1.53E-4	0.02701	3.17E-5	0.00511	0.00200	2.47E-4	0.01911	0.00886	0.00622	3.80E-5	0.00100	2.23E-4
6, 7	0.01714	0.01390	1.57E-4	0.03620	3.12E-4	0.00145	0.00552	0.03061	7.14E-4	0.01605	0.00945	2.96E-4	1.62E-5
6, 8	4.34E-5	0.00172	0.06659	0.00166	0.01947	0.00430	0.00331	7.84E-4	0.00771	0.00101	1.34E-4	4.42E-7	3.69E-4
6, 9	5.77E-5	0.03970	0.01903	2.92E-4	0.04756	0.03121	2.07E-6	3.68E-5	0.00212	3.27E-4	1.20E-4	1.90E-6	6.36E-6
6, 10	2.06E-4	0.00434	0.06997	0.00445	0.05882	0.00138	0.00303	3.41E-4	0.00229	0.00157	1.95E-4	5.31E-6	9.08E-4

requeridas en las *EPWavenets* es de $\bar{N} = 5.85$ con una desviación estándar de $\sigma_N = 2.85$. El error promedio de aproximación para los 13 casos fue $4.05E - 5$ con una desviación estándar de $\sigma = 6.74E - 5$.

Nótese que en el 69 % de los casos la mejor aproximación se obtuvo con filtros de longitud $L = 4$ y en el restante 31 % se obtuvo la mejor aproximación con filtros de longitud $L = 6$.

Esto se atribuye a la dificultad inherente de optimizar un parámetro β adicional, de hecho se puede demostrar que la parametrización con $L = 4$ es un subcaso de la parametrización de $L = 6$.

6.2. Experimento 2

Se comparó el resultado de la aproximación entre la mejor arquitectura *EPWavenet* del Experimento 1 y la aproximación de una RNFBR con función gaussiana (RNFBRG). Esto significa haber usado el mismo número de neuronas para cada $f\#$ pero diferente función base. Los resultados se muestran en la Tabla 3 en donde en la primera columna se muestra el tipo de red neuronal (*EPWavenet* o RNFBR) y en las columnas restantes se muestra el error de aproximación de cada una de las funciones $f1$ a $f13$. En la Tabla 3 se puede observar que en todos

Tabla 3: Comparación de *EPWavenets* y RNFBR con función gaussiana

Red neuronal	$f1$	$f2$	$f3$	$f4$	$f5$	$f6$	$f7$	$f8$	$f9$	$f10$	$f11$	$f12$	$f13$
EPWavenet	1.68E-8	1.53E-4	1.57E-4	3.17E-5	1.63E-4	2.91E-9	2.20E-10	8.87E-6	1.18E-5	1.79E-8	1.28E-16	4.42E-7	5.31E-8
RNFBRG	2.86E-5	7.86E-4	5.92E-4	1.96E-4	0.04887	9.01E-5	0.01233	2.17E-4	1.50E-4	1.85E-4	0.06237	8.45E-4	1.80E-6

los casos la *EPWavenet* mejora la aproximación de la RNFBRG. Se hace notar que el error promedio de la aproximación con *EPWavenets* es $4.04E-005$ con una desviación estándar de $6.74E-5$, en tanto que el error promedio de aproximación de la RNFBRG es $9.74E-3$ con una desviación estándar de $2.08E-2$.

6.3. Experimento 3

Se generalizó la ecuación (7) al introducir un desplazamiento $k \geq 0$ entre el i -ésimo dato de entrada en el momento m , y el i -ésimo dato de salida en el momento $m + k$ que matemáticamente se expresa así:

$$Error(k) = \frac{1}{M} \sum_{m=1}^{m=M} \sum_{i=1}^n (x_i(m) - y_i(m+k))^2 \quad (11)$$

Nótese que la ecuación (11) permite realizar la predicción de la función al establecer la dependencia $y(m+k) = f(x(m))$.

Se compararon las predicciones para $k = 1, \dots, 100^1$ para la mejor arquitectura *EPWavenet* (ver Experimento 1) y las RNFBRG con el mismo número de neuronas, con las funciones $f1$ a $f13$. Adicionalmente se consideró incluir una función sigmoïdal (que no es de base radial) para explorar su comportamiento como predictor usando también una capa oculta.

¹ A excepción del caso de $f12$ se usaron el 10% de las muestras, esto es $k = 1, 2, \dots, 28$.

Los resultados se muestran en la Figuras 4 y 5 mismas que en cada subfigura comparan a las *EPWavenets* con RNRBFG y redes neuronales con funciones sigmoidales.

En las Figuras 4 y 5 se puede apreciar que para f_1 , f_2 , y f_8 (Brownian Motion, Dow Jones y Lorenz) las *EPWavenets* tienen un mejor desempeño que el resto de las redes neuronales. También se observa que al incrementarse el valor de k el error de aproximación se incrementa paulatinamente.

Algo similar ocurre con la función f_9 (Rossler) en donde el error se incrementa conforme el valor de k , sin embargo la aproximación con *EPWavenets* se ve superada por el resto de las redes neuronales.

Los peores resultados con *EPWavenets* se dan con la función f_3 (EGC) que si bien es cierto ofrece la mejor aproximación con $k = 0$ falla con $k > 0$. En tanto que la RNFBRG y la red con función sigmoidal siguen un comportamiento suave e incremental.

Para los casos f_4 , f_5 , f_6 , f_7 y f_{13} (Henon, Ikeda, Laser, Logistic, y White Noise) el comportamiento es prácticamente el mismo para cualquier $k > 1$, es decir, sólo se puede aproximar para $k = 0$, y para $k > 0$ el error de aproximación se dispara de inmediato con cualquier tipo de red neuronal. De hecho la peor predicción se obtiene con la función sigmoidal que presenta saltos abruptos para diferentes valores de k , esto se atribuye a que no puede captar singularidades en las funciones objetivo y toda vez que este tipo de funciones siguen un comportamiento caótico.

Para los casos f_{10} , f_{11} y f_{12} (Seno, Seno con Ruido y Sun Spots) se observa que todas las redes neuronales se comportan de forma similar y que el valor de predicción fluctúa dependiendo de la periodicidad de las funciones. La peor predicción se logra con funciones sigmoidales, en tanto que la *EPWavenet* tiene un desempeño competitivo respecto a la RNFBRG.

7. Conclusiones

Se presentaron las *EPWavenets* que combinan técnicas de cómputo evolutivo, redes neuronales y wavelets paramétricos.

Los experimentos permiten afirmar que es posible aproximar funciones con *EPWavenets* inspirados en la arquitectura de redes neuronales de funciones de base radial con una sola capa oculta, y usar wavelets paramétricos sin requerir de su expresión analítica.

Las *EPWavenets* logran un alto grado de adaptabilidad y un desempeño competitivo respecto a otras redes neuronales que involucran funciones de base radial. Los experimentos nos permitieron comparar el desempeño de las *EPWavenets* con RNFBRG, y notamos que en cuanto a aproximación se refiere las *EPWavenets* son mejores en todos los casos.

En cuanto a predicción se refiere, al comparar la predicción de *EPWavenets* con RNFBR con funciones gaussianas y funciones sigmoidales (que no son de base radial) notamos que éstas últimas tienen un peor desempeño, lo cual se

atribuye a la no localización de la función sigmoïdal, y solo en un caso (uno de trece) las *EPWavenets* se ven consistentemente superadas.

Adicionalmente notamos que se requiere un número reducido de neuronas con wavelets para realizar la tarea de aproximación ($\bar{N} = 5.85$) lo cual se atribuye al uso de *wavenets* paramétricas ortogonales que minimizan el número de funciones wavelet requeridas al ser bases no redundantes.

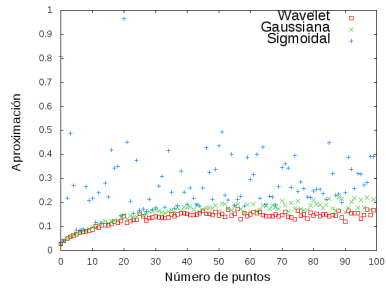
8. Trabajo futuro

Se tiene contemplado extender los experimentos para hacer comparaciones con otras FBR como son la función cuadrática, la función multicuadrática, la función inversa cuadrática, y las funciones *splines*. Además de usar parametrizaciones de mayor longitud, y plantear el problema de aproximación con *EPWavenets* como un problema multiobjetivo. También se contempla incrementar la dimensión n de los datos de entrada, toda vez que en este artículo se trabajó con vectores de entrada unidimensionales. Otros trabajos futuros también incluyen aplicar *EPWavenets* en problemas de clasificación y reconocimiento de patrones.

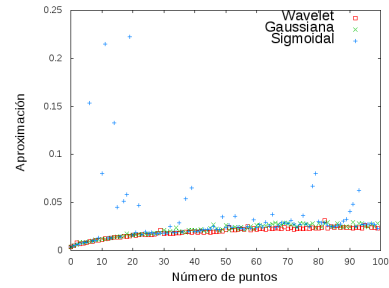
Referencias

1. Cybenko, G.: Approximation by Superpositions of a Sigmoidal Function. *Mathematics of Control, Signals, and Systems* pp. 303–314 (1989)
2. Daubechies, I.: Ten lectures on wavelets. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA (1992)
3. Delyon, B., Juditsky, A., Benveniste, A.: Accuracy analysis for wavelet approximations. *Neural Networks, IEEE Transactions on* 6(2), 332–348 (Mar 1995)
4. Haykin, S.: *By Simon Haykin Neural Networks: A Comprehensive Foundation*. Prentice Hall, 2 edn. (Jul 1998)
5. Herrera, O., Mora, R.: Aplicación de Algoritmos Genéticos a la Compresión de Imágenes con Evolets. *Sociedad Mexicana de Inteligencia Artificial* (2011)
6. Kononov, E.: Visual Recurrence Analysis (Apr 2015), http://www2.informatik.uni-osnabrueck.de/marc/lectures/zra_ss03/prgdat/vra4v2.zip
7. Mallat, S.G.: A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Trans. on Patt. Anal. Mach. Intell.* 11(7), 674–693 (1989)
8. Mallat, S.G.: Multiresolution approximations and wavelet orthonormal bases of $L^2(\mathbb{R})$. *Trans. Amer. Math. Soc.* 315(1), 69–87 (1989)
9. Pourtaghi, A.: Wavelet Neural Network and Wavenet Performance Evaluation in Hydrodynamic Force Prediction due to Waves on Vertical Cylinders. *International Journal of Information and Computer Science* 11(9), 187–213 (dec 2012)
10. Roach, D.W., Lai, M.: Parameterizations of univariate orthogonal wavelets with short support, pp. 369–384. *Vanderbilt Univ. Press* (2002)
11. Sotelo, C.: The chemotactic hypothesis of Cajal: a century behind. *Progress in brain research* 136, 11–20 (2002)
12. Tewfik, A., Sinha, D., Jorgensen, P.: On the Optimal Choice of a Wavelet for Signal Representation. *IEEE Transactions on information theory* 38(2), 747–765 (march 1992)

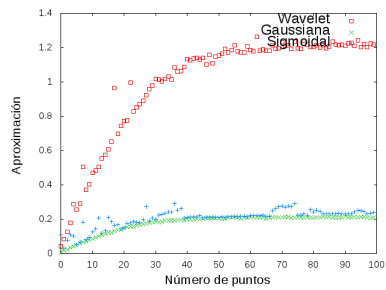
13. Ypma, A., Duin, R.P.: Using the Wavenet for function approximation pp. 23–6 (1997)
14. Zhang, Q., Benveniste, A.: Wavelet Networks. *IEEE Trans. Neural Networks* 3(6), 889–898 (1992)



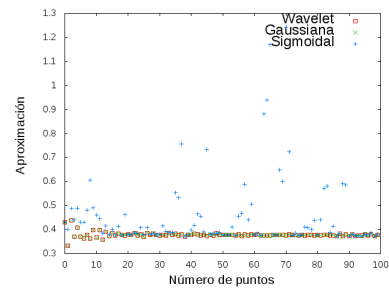
(a) Predicción de f_1 con 8 neuronas



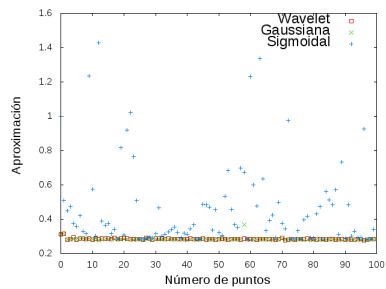
(b) Predicción de f_2 con 6 neuronas



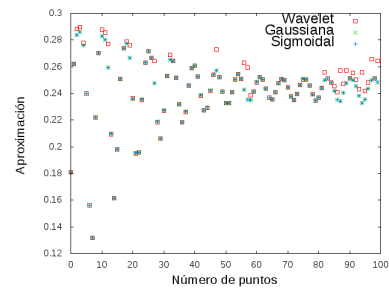
(c) Predicción de f_3 con 7 neuronas



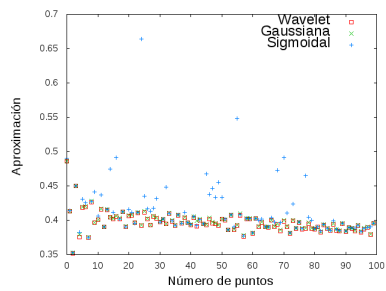
(d) Predicción de f_4 con 6 neuronas



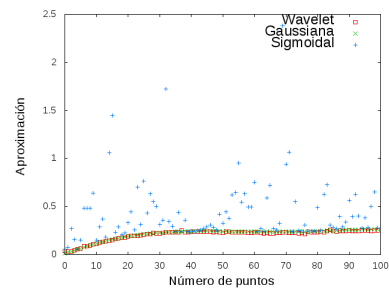
(e) Predicción de f_5 con 8 neuronas



(f) Predicción de f_6 con 1 neurona

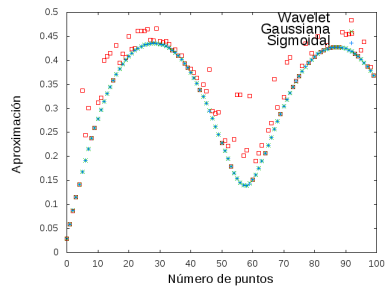


(g) Predicción de f_7 con 7 neuronas

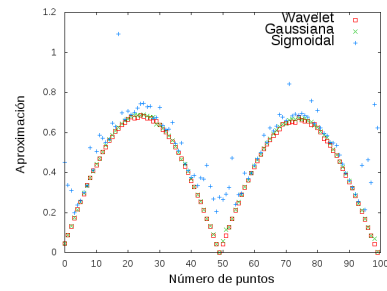


(h) Predicción de f_8 con 8 neuronas

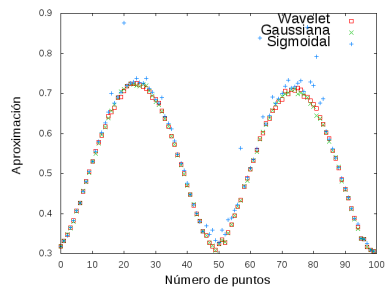
Fig. 4: Predicción de funciones f_1 a f_6



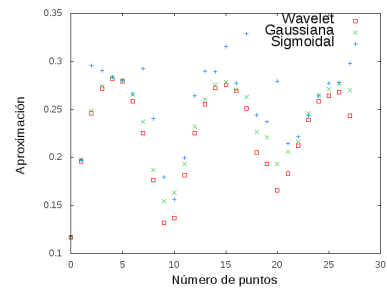
(a) Predicción de f_9 con 1 neurona



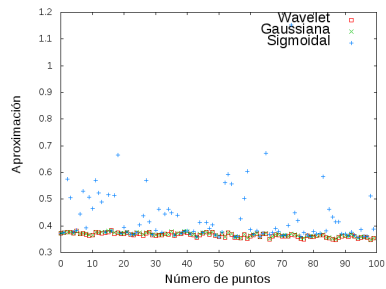
(b) Predicción de f_{10} con 8 neuronas



(c) Predicción de f_{11} con 1 neurona



(d) Predicción de f_{12} con 8 neuronas



(e) Predicción de f_{13} con 8 neuronas

Fig. 5: Predicción de funciones f_7 a f_{13}

Hacia la construcción de una herramienta para la creación de bitácoras para Android

Bárbara Cervantes¹, Miguel González-Mendoza¹, Raúl Monroy¹, Christian Eduardo Galdámez Blanco², Eduardo Ismael García Pérez²

¹ Tecnológico de Monterrey, Campus Estado de México, México

² Universidad Politécnica de Chiapas, México

{bcervantesg,mgonza,raulm}@itesm.mx
{christianeduardogb,eduardo78d}@gmail.com
<http://www.itesm.mx>
<http://www.upchiapas.edu.mx>

Resumen. El uso de teléfonos inteligentes se ha vuelto muy popular. La interacción de los usuarios con estos dispositivos genera información que describe su comportamiento en el sistema. Sistemas operativos como Android [17], ofrecen la posibilidad de registrar muchos aspectos de esta interacción, sin embargo, esta información generalmente es utilizada en el momento en el que sucede la interacción sin que se cree un registro persistente de todo lo ocurrido. El acceso a esta información puede resultar muy útil para la caracterización del usuario, un área con múltiples aplicaciones. En este documento se presenta el avance realizado hacia la construcción de una herramienta que permita la colección de información del uso de un dispositivo móvil para su posterior análisis y aprovechamiento.

Palabras clave: interacción humano-computadora, comportamiento del usuario, caracterización del usuario, Android.

1. Introducción

Avances en la tecnología nos han llevado a un mundo en el que la adopción de los dispositivos móviles se ha expandido considerablemente. A diferencia de hace diez años, cuando los teléfonos móviles únicamente contenían el historial de llamadas, algunos contactos y mensajes SMS; ahora, cada dispositivo contiene información acerca de la vida diaria de la persona. El surgimiento de aplicaciones de índoles muy diversos (email, redes sociales, entretenimiento, productividad, salud, etc.) ha potenciado el uso de teléfonos inteligentes y consecuentemente se ha incrementado la cantidad de información generada dentro de los mismos. Cada uno de estos dispositivos está fuertemente ligado a la identidad de la persona y, además, contiene información personal que puede llegar a ser sensible por lo que se debe tener presente la seguridad de esta información.

Cuando un usuario interactúa con un sistema computacional, en este caso móvil, genera información que describe su comportamiento en el sistema. Esta

información puede ser extraída y representada en un modelo a través del cual se pueda reconocer al usuario. Nuestro trabajo se centra en recopilar la información generada por el usuario para su posterior análisis y aprovechamiento. En particular, para este primer avance, se lleva un registro de las tareas que se ejecutan en un dispositivo Android; estas tareas (definidas por el sistema operativo) describen la forma en que se navega a través del dispositivo y las aplicaciones, por lo que las consideramos parte importante de la interacción entre el usuario y el dispositivo móvil.

Existen muchas posibles aplicaciones que se benefician o se basan en conocer el comportamiento de los usuarios. Entre las áreas de aplicación tradicionales de modelado de usuario se encuentran la personalización de servicios [14], los sistemas de recomendación [11,8,14], la mejora de calidad de servicios [12], etc. Áreas de aplicación más recientes y más orientadas a dispositivos móviles incluyen el monitoreo de la salud [13], caracterización de actividades [7,10] y mejoras en seguridad [6,9,16,15], entre otras. Un enfoque es la creación de un modelo de usuario que describa las características relevantes para el problema en cuestión [6,8,9]; otro es la abstracción de un modelo de comportamiento general a partir del cuál se trata de inferir el comportamiento de un usuario en específico [12,10].

En este documento se reporta el avance realizado hacia la construcción de una herramienta que permita la colección de información del uso de un dispositivo móvil para su posterior análisis. Para esto comenzamos por dar una visión general del proceso de creación de bitácoras, seguido del caso específico de sistemas móviles (Sección 2). Posteriormente, en la sección de Arquitectura (Sección 3), describimos los componentes principales de la herramienta y cómo se relacionan, se mencionan las decisiones de diseño tomadas y algunas pruebas realizadas. Lo que nos lleva a presentar el estado en el que se encuentra la herramienta y el trabajo que se realizará en el marco de este proyecto (Sección 4). Finalmente viene la sección de Conclusiones (Sección 5), en la que incluimos nuestros comentarios finales.

2. Trabajo relacionado

Una bitácora (o historial) es un archivo en el que se registran los eventos que ocurren en un sistema operativo, aplicación o algún proceso que se quiera monitorear o respaldar. Este tipo de archivos son generalmente utilizados para entender las actividades del sistema. En el caso de los sistemas operativos para servidores o computadoras personales, existen herramientas que permiten activar el registro de eventos en una bitácora; por ejemplo: *Linux Audit framework* [18] es una herramienta que permite registrar desde la ejecución de archivos hasta las llamadas a sistema. El módulo del kernel *audit* intercepta las llamadas a sistema y guarda los eventos relevantes; el demonio *auditd* escribe estos registros en el disco y varias herramientas de línea de comando (*autrace*, *ausearch* y *aureport*) sirven para visualizar los eventos registrados en la bitácora (ver Figura 1).

Event Summary Report	
total	type
2434	SYSCALL
816	USER_START
816	USER_ACCT
814	CRED_ACQ
810	LOGIN
806	CRED_DISP
779	USER_END
99	CONFIG_CHANGE
52	USER_LOGIN

Fig. 1. Ejemplo de reporte de una bitácora, usando *aureport* (tomado de [18]).

En cambio, cuando hablamos de dispositivos móviles, la mayoría de las herramientas que permiten ver los eventos que ocurren en el sistema están enfocadas a la etapa de desarrollo de una aplicación, cuando se activa el modo de depuración, o a ver el estatus del dispositivo en el tiempo actual. Esto se debe en parte a que el proceso de colección de datos para una bitácora viene a costa del rendimiento. Los dispositivos móviles son equipos más limitados en este sentido (al tener un espacio de almacenamiento mucho menor que el de una PC y mayores restricciones de energía) por lo que es imperativo que las herramientas de colección de datos sean optimizadas para no afectar el funcionamiento y experiencia de uso del móvil.

La mayoría de las aplicaciones disponibles para observar las tareas que se ejecutan en un sistema Android, simplemente consisten en una interfaz gráfica que muestra la información del momento actual al usuario, sin embargo, no existe la función del historial, es decir no hay un registro como tal (ver Figura 2).

También existen sistemas más completos que atacan un problema en específico, por ejemplo, *TaintDroid* [4] es una extensión a Android con el objetivo de comprender el flujo de los datos para identificar aplicaciones que traten la información sensible de manera inapropiada. Otro ejemplo, dedicado a la creación de bitácoras se expone en [5], en este caso el problema que se abarca es la detección de fallas en el sistema.

El sistema Android cuenta con una herramienta de registro de mensajes *LogCat* [21], sin embargo, como se mencionó anteriormente, está pensada más bien con fines de depuración durante el desarrollo de la aplicación. La inclusión de un nuevo registro se programa dentro de cada aplicación por lo que cada mensaje tiene la estructura que el programador decida, lo cual dificulta la interpretación de cada uno de estos mensajes. El tipo y la cantidad de mensajes en LogCat son muchos y normalmente se observan en el IDE del desarrollador a través de una conexión con cable USB, en situaciones normales, es decir, cuando el teléfono es utilizado por el usuario final, debido a las limitaciones de espacio y procesamiento del teléfono móvil la bitácora se sobrescribe. Crear una bitácora a partir de lo registrado con LogCat no resulta práctico debido a que es difícil determinar el momento en el que se sobrescribirá el archivo y además habría que

estar filtrando este archivo para obtener únicamente los eventos relevantes, por esta razón vemos la necesidad de crear una herramienta que registre los eventos que nos interesan, como es el caso de [5].



The screenshot shows a mobile application interface with a dark theme. At the top, there are icons for a signal strength indicator, an information icon, a filter icon, and a close icon. Below these is a table with four columns: PROCESS, CONNECTION, MISC, and MEM. The table lists various system and user processes with their IDs, names, and CPU usage percentages. At the bottom, it shows memory usage statistics: Memory Total: 797M and Free: 288M.

PROCESS	CONNECTION	MISC	MEM
Process: 43		CPU Usage: 11.8%	
0	System		4.8
9452	OS Monitor		2.3
460	Android System		2.1
7648	OS Monitor		0.9
721	Phone		0.5
588	System UI		0.4
9471	OS Monitor		0.4
845	Google Services Framework		0.2
1180	Maps		0.2
2623	Messaging		0.2
149	/sbin/adbd		0.0
711	Google Pinyin		0.0
743	Launcher		0.0
808	com.android.smsush		0.0
Memory Total: 797M		Free: 288M	

Fig. 2. Ejemplo de una aplicación que permite ver el estado del dispositivo, OS Monitor [20], se muestran los procesos que corren actualmente y detalles del uso de CPU y la memoria, sin embargo, no hay un registro persistente de estos datos.

3. Arquitectura

La herramienta tiene como objetivo principal registrar los eventos que describen la interacción del usuario con el dispositivo móvil. Entre estos eventos están las interacciones con la pantalla, lecturas de sensores como el acelerómetro o giroscopio y las aplicaciones ejecutadas. Para la primera etapa del proyecto se decidió incluir en la bitácora las tareas y procesos que se ejecutan en un dispositivo Android; éstos describen la forma en que se navega a través del

dispositivo y las aplicaciones, por lo que las consideramos parte importante de la interacción entre el usuario y el dispositivo móvil, y creemos que serán un buen punto de partida para nuestra investigación. Además tienen la ventaja de poder ser obtenidas a bajo costo y estar siempre presentes, es decir, no dependen de la conectividad de la red o de que suceda un evento en específico y no requieren prender ningún sensor.

En esta Sección se describen los componentes principales de la herramienta y cómo se relacionan. Empezamos por mencionar algunas decisiones de diseño y posteriormente describimos cada elemento de la arquitectura.

3.1. Antecedentes

En esta Sección se describen algunas de las generalidades de Android y se mencionan las decisiones de diseño derivadas de ellas:

Almacenamiento

Existen dos alternativas en cuanto a dónde almacenar la bitácora: usar el almacenamiento externo del dispositivo, comúnmente una tarjeta SD, o usar el almacenamiento interno, menor en tamaño pero siempre disponible. Descartamos la posibilidad de utilizar el almacenamiento externo porque a pesar de que se tendría menor restricción en cuanto a espacio de almacenamiento, esta memoria puede ser leída y modificada por el usuario y por otras aplicaciones. Se eligió guardar la bitácora en el almacenamiento interno del dispositivo ya que un archivo almacenado en la memoria interna del dispositivo solamente puede ser accedido por la aplicación a la que pertenece, para otras aplicaciones e incluso para el dueño del dispositivo está restringido el acceso de estos archivos

En cuanto al formato, se eligió almacenar los registros en una base de datos. La razón principal es nuevamente que una base de datos no será accesible para el usuario u otras aplicaciones. Android provee un soporte de SQLite, [22]. SQLite permite a cualquier aplicación nativa de Android el acceso al almacenamiento en bases de datos, se provee una forma fácil de hacer la conexión y operaciones simples como consultas, inserciones, actualizaciones y borrado de registros; así como operaciones más complejas como uniones, pivotes y otras más. Cada uno de los registros serán almacenados en una base de datos que estará en constante actividad ya que el recabado de registros será constante y por lo consiguiente la inserción también.

Ejecución

Se decidió implementar la herramienta como un servicio del sistema operativo. Un servicio (*android.app.Service* [19]) es un componente de una aplicación que puede realizar operaciones de larga duración en segundo plano y que puede tener, o no, una interfaz de usuario. Un servicio puede continuar ejecutándose en segundo plano incluso si el usuario cambia a otra aplicación.

3.2. Componentes

La herramienta tiene dos componentes principales: un colector y un servidor. El colector reside en el dispositivo móvil y es el encargado de registrar los eventos en la bitácora, la cual se transfiere periódicamente al servidor en donde se almacena permanentemente. A continuación se describen los detalles de implementación de cada uno de los componentes.

Colector

El Colector es quien se encarga de insertar en la base de datos las nuevas tareas (*Android tasks*). Para implementarlo se siguió el paradigma de Modelo-Vista-Controlador (MVC) [2], cada uno de los elementos se encuentra en un paquete diferente. Se siguió este paradigma para tener un diseño modular y que el hecho de registrar un nuevo tipo de evento en la bitácora no involucre modificar toda la arquitectura.

En el primer paquete, llamado «Controllers», se encuentran los controladores; dentro de él tenemos el mayor número de clases las cuales permiten recabar todas las tareas que el dispositivo genere. Es aquí donde se define qué eventos deben ser monitoreados. Como se ha mencionado anteriormente, por el momento obtenemos los procesos y tareas del sistema operativo; esto se logra a través de la clase *ActivityManager* [23]. Este paquete también se encarga de iniciar la transferencia de la bitácora al servidor y de tomar las medidas adecuadas (iniciar o detener el registro de eventos) cuando el dispositivo cambia de estado.

En el paquete «Models» podemos encontrar las clases que crean y manipulan la base de datos. Aquí es donde se definen los modelos de los eventos por obtener: tenemos el caso de *AndroidProcess* y *AndroidTask* para los que se obtienen el nombre de la tarea, identificadores y la fecha (Figura 3). Cada evento es un registro en la base de datos, los cuales se añaden secuencialmente.

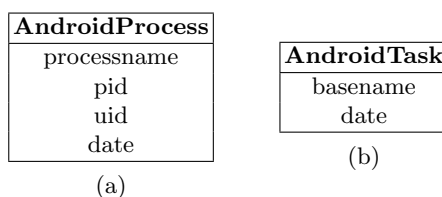


Fig. 3. Cada evento tiene diferentes características a registrar, en el modelo del evento se definen estas características. En la imagen se observa el modelo para los procesos(a) y tareas(b) de Android.

Por último el paquete «Views» contiene las clases necesarias para el funcionamiento de la interfaz gráfica. Aunque la interacción con el usuario es mínima,

existe una interfaz gráfica que permite al usuario confirmar si se llevará a cabo la transmisión de la bitácora al servidor (para evitar realizarla en un momento no deseado) y elegir el tipo de eventos a registrar.

A grandes rasgos el funcionamiento del Colector es el siguiente: al iniciar el dispositivo se crea un thread de servicio que estará monitoreando las tareas del sistema operativo; en el momento en que se detecta una tarea nueva se crea el modelo que la representa y se añade a la base de datos. El colector también está pendiente de los cambios de estado del dispositivo, es decir, cuando pasa a un estado suspendido, cuando sale de este mismo estado, cuando se reinicia, etc., y realiza las acciones pertinentes para que la colección siga activa sin necesidad de interacción por parte del usuario del dispositivo. Existe la posibilidad de detener el proceso de registro si el usuario así lo desea y de elegir el tipo de información que se registra. Por otro lado tiene una alarma programada para que cada día se intente hacer la transmisión al servidor, esta alarma se activa a la hora especificada y pregunta al usuario si desea hacer la transmisión en ese momento, en caso de decidir no hacerlo se enviará un recordatorio a los 30 minutos.

Servidor

El servidor recibe los archivos del historial para realizar un respaldo, el objetivo es que se pueda liberar ese espacio del dispositivo móvil y que la bitácora quede almacenada en un servidor seguro para su posterior análisis.

Para crear el servidor se utilizó el lenguaje de programación Java debido a que se desean heredar las características de seguridad que ofrece. El servidor es de tipo *socket* (*ServerSocket*), el cual está diseñado para estar siempre a la escucha de peticiones de los clientes para realizar el respaldo. Se define un puerto que es el que escuchará las peticiones. El servidor tiene la capacidad de crear subprocesos concurrentes para poder realizar el respaldo de más de una bitácora al mismo tiempo; con ayuda de la clase *Thread* es posible crear n número de subprocesos dentro del servidor los cuales cobran vida cada vez que una petición es aceptada por el servidor, de esta forma se evita la sobrecarga del servidor.

3.3. Seguridad

Debido a que la información que se maneja en las bitácoras es información sensible, buscamos que al hacer la transmisión la información no se vea comprometida. Para esto se establece una conexión entre el dispositivo y el servidor siguiendo el protocolo de Diffie-Hellman[1].

Otra de las medidas tomadas para asegurar la transmisión de una bitácora del dispositivo al servidor es que se realice en forma cifrada. Java en conjunto con el paquete de seguridad «*security*» hace uso del algoritmo AES[3] para el cifrado de los datos que el cliente, en este caso el dispositivo móvil, esté enviando por medio de la red. Uno de los puntos a tomar en cuenta dentro de este algoritmo

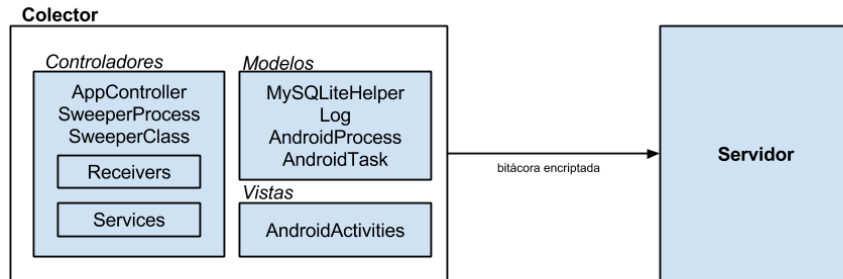


Fig. 4. Arquitectura del sistema

es que ya que es un algoritmo que se centra en el cifrado por bloques, contiene un límite de bits, en este caso se trabaja con 16 bits. Los datos se envían por partes y cifrados hacia el servidor el cual una vez aceptando la petición del cliente comienza a recibir los bloques de 16 bits desde el cliente y descifra estos datos. Una vez descifrados los datos pasan a ser concatenados en el servidor para que cuando la transmisión de datos se haya completado el archivo quede almacenado sin ningún problema y sin ningún byte perdido en la transferencia de los datos por la red.

3.4. Pruebas

Hasta el momento se han realizado pruebas en dos teléfonos móviles. El objetivo de estas primeras pruebas era comprobar que el hecho de tener la herramienta corriendo no afecta la experiencia de usuario (desempeño) del dispositivo y medir el espacio que las bitácoras ocupan en el dispositivo, ya que al estar almacenadas en la memoria interna si fueran de gran tamaño causarían inconveniente al usuario. Después de instalar la herramienta y utilizar el celular de manera cotidiana los dos usuarios reportaron no haber percibido diferencia en el desempeño del dispositivo, el sistema corría a la misma velocidad y el consumo de batería no fue notorio, y después de un día de interacción el tamaño de la bitácora no superó los 64 kB. En pruebas posteriores es nuestra intención formalizar estas percepciones, es decir, determinar exactamente cuál es el porcentaje de energía que se dedica a la herramienta y cómo varía el tamaño de la bitácora dependiendo del usuario y los eventos registrados.

4. Estado actual y trabajo futuro

Hasta el momento la herramienta tiene la opción de registrar dos tipos de eventos: procesos y tareas del sistema operativo. Se busca extenderla para registrar otros tipos de eventos, como ejemplo tenemos el uso de sensores del

propio dispositivo (*i.e.* giroscopio, GPS, etc.) así como de interfaz con el usuario (*i.e.* touch screen).

Se busca desplegar el servidor y reclutar a un grupo piloto de usuarios para la creación de una base de datos de interacción con teléfonos móviles. Una vez conformada, se pretende dar un enfoque hacia la identificación de los niveles de interacción entre el dispositivo y el usuario, enfocado hacia la detección de anomalías. Se evaluará el uso de estos dos tipos de eventos en la autenticación implícita[9], se espera que los resultados sean positivos bajo la hipótesis de que éstos son capaces de capturar la interacción que distingue a un usuario de otro. Además, buscamos considerar que la manera en que interactúa el usuario depende del nivel de carga mental que tiene en ese momento, es decir, la interacción no será la misma cuando el usuario presta 100 % de su atención al dispositivo que cuando el usuario está caminando o realizando alguna otra actividad en paralelo al utilizar el dispositivo móvil, probablemente en este último escenario el tiempo entre las tareas que ejecuta el usuario sea mayor. Esto se deberá de ver reflejado en el modelo del comportamiento de cada usuario.

Posteriormente, se extendería la herramienta para que esta colección de bitácoras no solamente sirva para crear una base de datos sino que además tenga como objetivo monitorear en tiempo real el uso del dispositivo para identificar cuando el usuario salga de su modelo de comportamiento.

5. Conclusiones

La versión aquí presentada de la herramienta para el registro de bitácoras en un sistema operativo para dispositivos móviles (Android), permite el registro de las tareas y los procesos del sistema operativo que el usuario ejecuta deliberadamente. Posteriormente se registrarán también eventos resultantes del uso de sensores del dispositivo.

Esta herramienta nos ayuda a sobrepasar el obstáculo tecnológico que implica la colección de las bitácoras para poder enfocarse en las aplicaciones que nacen del análisis del comportamiento de los usuarios.

De igual forma, amplía el horizonte de estudio para adaptar a este paradigma en móviles (dadas las restricciones de capacidad de cómputo y energéticas), estudios de análisis forense y seguridad de entornos de cómputo tradicionales (con mayor capacidad de procesamiento, almacenaje y sin restricciones energéticas importantes).

Referencias

1. Diffie, W., Hellman, M. E.: New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6), 644–654 (1976)
2. Krasner, G. E., Pope, S. T.: A Description of the Model-View-Controller User Interface Paradigm in the Smalltalk-80 System. (1988)
3. Daemen, J., Rijmen, V.: The design of Rijndael: AES-the advanced encryption standard. Springer Science & Business Media (2002)

4. Enck, W., Gilbert, P., Chun, B.-G., Cox, L. P., Jung, J., McDaniel, P., Sheth, A. N.: TaintDroid: An Information Flow Tracking System for Real-time Privacy Monitoring on Smartphones. *Commun. ACM*, 57(3), 99106 (2014)
5. Cinque, M., Cotroneo, D., Testa, A.: A Logging Framework for the On-line Failure Analysis of Android Smart Phones. In: *Proceedings of the 1st European Workshop on AppRoaches to MObiquiTous Resilience*, Vol. 1, pp. 2:12:6, ACM (2012)
6. Schmidt, A. D., Peters, F., Lamour, F., Scheel, C., Çamtepe, S. A., Albayrak, Ş: Monitoring Smartphones for Anomaly Detection. *Mobile Networks and Applications*, 14(1), 92106 (2009)
7. Cao, H., Bao, T., Yang, Q., Chen, E., Tian, J.: An Effective Approach for Mining Mobile User Habits. In: *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, pp. 16771680, New York, NY, USA: ACM (2010)
8. Woerndl, W., Schueller, C., Wojtech, R.: A Hybrid Recommender System for Context-aware Recommendations of Mobile Applications. In: *Data Engineering Workshop, 2007 IEEE 23rd International Conference on*, pp. 871878 (2007)
9. Shi, E., Niu, Y., Jakobsson, M., Chow, R.: Implicit authentication through learning user behavior. *Lecture Notes in Computer Science*, vol. 6531, pp. 99113, Springer-Verlag Berlin Heidelberg (2011)
10. Furlletti, B., Gabrielli, L., Renso, C., Rinzivillo, S.: Identifying Users Profiles from Mobile Calls Habits. In: *Proceedings of the ACM SIGKDD International Workshop on Urban Computing*, pp. 1724, New York, NY, USA: ACM (2012)
11. Park, S., Kang, S., Kim, Y.-K.: A channel recommendation system in mobile environment. *IEEE Transactions on Consumer Electronics*, 52(1), 3339 (2006)
12. Wu, S., Fan, H.-H.: Activity-Based Proactive Data Management in Mobile Environments. *IEEE Transactions on Mobile Computing*, 9(3), 390404 (2010)
13. Chan, V., Ray, P., Parameswaran, N.: Mobile e-Health monitoring: an agent-based approach. *IET Communications*, 2(2), 223230 (2008)
14. Gavalas, D., Kenteris, M.: A web-based pervasive recommendation system for mobile tourist guides. *Personal and Ubiquitous Computing*, 15(7), 759770 (2011)
15. Frank, M., Biedert, R., Ma, E., Martinovic, I., Song, D.: Touchalytics: On the Applicability of Touchscreen Input as a Behavioral Biometric for Continuous Authentication. *IEEE Transactions on Information Forensics and Security*, 8(1), 136148 (2013)
16. Feng, T., Yang, J., Yan, Z., Munguia Tapia, E., Shi, W.: TIPS: Context-aware Implicit User Identification Using Touch Screen in Uncontrolled Environments. In: *Proceedings of the 15th Workshop on Mobile Computing Systems and Applications (HotMobile14)*, pp. 9:19:6, ACM (2014)
17. Google, Android, <http://www.android.com/>
18. The Linux Audit Framework, https://www.suse.com/documentation/sled10/pdf/doc/audit_sp1/audit_sp1.pdf
19. Google, Android Service, <http://developer.android.com/reference/android/app/Service.html>
20. OS Monitor - Android Apps on Google Play, <https://play.google.com/store/apps/details?id=com.eolwral.osmonitor>
21. Google, LogCat, <http://developer.android.com/tools/help/logcat.html>
22. SQLite, <https://www.sqlite.org/>
23. Google, Activity Manager, <http://developer.android.com/reference/android/app/ActivityManager.html>

Belleza artificial: evolucionando partes del rostro

Ricardo Solano Monje¹, Nayeli Joaquinita Meléndez Acosta², Sergio Juárez Vázquez¹, Homero Vladimir Ríos Figueroa³

¹ Universidad del Istmo, Campus Sto. Domingo Tehuantepec, Oax., México

² Universidad del Istmo, Campus Ixtepec, Oax., México

³ Universidad Veracruzana, Xalapa, Ver., México

rsolanomonje@gmail.com, sjuarez@bianni.unistmo.edu.mx,
nayelimelendez@gmail.com, hrios@uv.com.mx
<https://www.uv.mx/dia/>

Resumen. Este artículo se enfoca en el desarrollo de una maquina embellecedora facial - algoritmo para evolucionar la máscara del rostro de entrada y embellecerlo-que opera como herramienta automatizada de retoque. El aspecto más importante que se considera en este artículo es el de enfocar. La idea de enfocar es identificar las características faciales que necesitan una mejora y concentrar el método de optimización únicamente sobre estas. Entonces la complejidad de manejar un sub-problema, en general, es más pequeña que tratar con el problema entero. Enfocar es implementado para permitir al usuario elegir las características a embellecer. Este algoritmo evolutivo utiliza un modelo SVR, entrenado anticipadamente usando la calificación de belleza generado por Leyvand. El algoritmo nos da la posibilidad de obtener varias “versiones del YO original”, algunas más embellecidas que otras. Hemos usado la base de datos Fg-Net (Face and Gesture Recognition Research Network).

Palabras clave: algoritmo evolutivo, embellecimiento artificial del rostro, aprendizaje automático.

1. Introducción

El embellecimiento artificial tiene sustento en estudios científicos sobre psicología social y ciencias de la psicología.

En [1] concluyen que son suficientes unos pocos milisegundos para crear una buena o mala impresión sobre una persona. Por otro lado, se sabe que la mente guarda o modela el rostro con rostros prototipo, los cuales son burdos o son carentes de detalles, de forma que se pueden corregir o complementar en la realidad [2]. La mente por

consiguiente, reconoce los rostros con información de bulto o sin detalles. Esto significa que los detalles suelen ser omitidos la primera vez que se ve un rostro, o compensados cuando se ve el rostro por segunda vez. Es decir, que por analogía al reconocimiento de rostros, el embellecimiento artificial pasaría desapercibido para el observador hasta cierto punto [3], [4].

Este artículo presenta un algoritmo evolutivo para el embellecimiento artificial del rostro completo o enfocado a cierta área del rostro. El contenido del artículo está organizado en seis secciones de la siguiente manera: la segunda sección trata sobre cómo se ha tratado el problema del embellecimiento digital en algunos trabajos relacionados. En la tercera sección se muestra el proceso de embellecimiento general en este artículo. En la cuarta sección se explica el funcionamiento del algoritmo evolutivo utilizado en este trabajo. La quinta sección muestra los resultados obtenidos utilizando nuestro método de embellecimiento. Y finalmente en la sexta sección se muestran las conclusiones.

2. Embellecimiento del rostro

Un grupo de investigadores en [4] crearon un modelo de aprendizaje automático capaz de calificar la belleza de rostros, dando resultados semejantes a los dados por jueces humanos. Es entonces que Leyvand en [5] propone la técnica para embellecer rostros haciendo uso de dicho modelo de aprendizaje automático.

En [5] para realizar el embellecimiento digital, se parte de la imagen facial de entrada y se determina un conjunto de características faciales predefinidas. Las características faciales -8 en total- están formadas por puntos de referencia y corresponden a 2 cejas, 2 ojos, labio superior, labio inferior, contorno de la nariz y contorno del rostro. A partir de las características faciales y de la posición de los puntos de referencia se calcula un conjunto de distancias entre dichas características.

Leyvand en [5] localiza 84 puntos de referencia, mientras que Eisenthal en [4] localiza 37 puntos de referencia. Con estos puntos se construye un vector \mathbf{v} una máscara. El objetivo de la máscara es formar un vector de distancias o longitudes entre los puntos de referencia que definen las características faciales. Leyvand obtiene 234 distancias. La diferencia de localización de puntos entre estos dos trabajos tiene una explicación sencilla, Leyvand usa la máquina calificadora de Eisenthal, pero mientras que Eisenthal persigue en su trabajo calificar la belleza de un rostro, Leyvand realiza la modificación y embellecimiento de un rostro.

3. Embellecimiento evolutivo

El principal componente de esta investigación es la máquina embellecedora - algoritmo para evolucionar la máscara del rostro de entrada y embellecerlo-, que hace uso de un conjunto de imágenes de rostros de mujeres -el ensamble de entrenamiento y sus calificaciones de belleza asociadas. El proceso de embellecimiento se muestra en la Fig. 1.

Dada una foto frontal como entrada donde el usuario ya seleccionó las características faciales a modificar –aunque podría ser todo el rostro de forma automática- se identifican en primera instancia los puntos de referencia que conllevan a especificar varias características faciales.

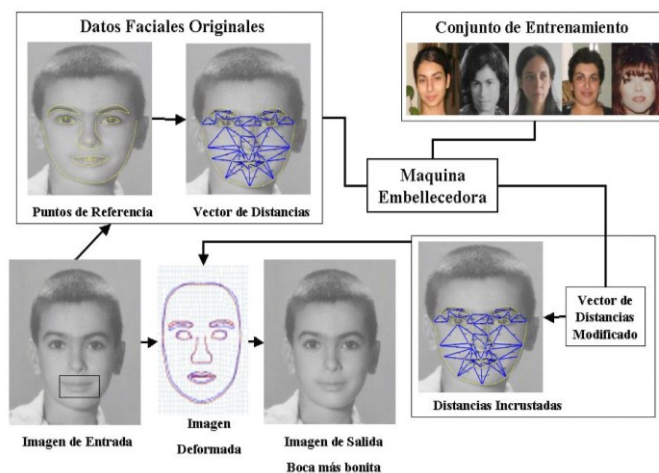


Fig. 1 Procedimiento de Embellecimiento

A partir de los puntos de referencia y entre las características faciales, se extrae un vector de distancias, que corresponde a la malla o máscara del rostro de entrada. Esta máscara alimenta al Algoritmo Evolutivo, el cual produce un vector de distancias modificado, que posee una calificación de belleza más alta que la del vector original.

La modificación de la imagen de entrada se llevará a cabo con transformaciones warping, este proceso mapea el conjunto de puntos de referencia $\{p_i\}$ de la imagen origen en el correspondiente conjunto de puntos de referencia $\{q_i\}$ de destino. De la Fig. 1 recuérdese que la máscara -el vector de distancias- permite realizar la transformación warping del rostro de entrada para producir el rostro de salida embellecido.

ENFOCAR. La idea de enfocar es identificar las características faciales que necesitan una mejora y concentrar el método de optimización únicamente sobre estas. Entonces la complejidad de manejar un sub-problema, en general, es más pequeña que tratar con el problema entero. Enfocar es implementado para permitir al usuario elegir las características a embellecer. Los puntos de referencia de las características elegidas son llamados puntos seleccionados, el resto son llamados puntos fijos. Con esto, el método de optimización se enfoca en embellecer las características faciales seleccionadas, mientras los puntos de referencia fijos preservan sus coordenadas espaciales. La Fig. 2 muestra una imagen de entrada donde se elige la boca.

Enfocar no es un proceso completamente automático, se considera una aplicación interactiva donde el usuario juega un rol activo para ayudar en el método de optimización -en realidad al proceso de afinar el embellecimiento a la medida- en la cual el usuario selecciona una o varias de las características faciales del rostro de entrada. Al interactuar con el método de optimización el usuario proporciona la o las características a embellecer, este ajuste reduce el espacio de soluciones a ser explorado, pues de

no especificar características faciales específicas, el rostro es embellecido en su totalidad.



Fig. 2 Rostro a embellecer con una característica facial seleccionada –boca

4 Evolucionando la máscara

En nuestro caso y para fines explicativos, se considera que partiendo del rostro de entrada se extrae un total de 66 puntos de referencia. Al igual que [5], los puntos de referencia son localizados en los contornos de las 8 diferentes características faciales: las dos cejas, los dos ojos, el labio inferior y superior de la boca, la nariz y límites del rostro, Fig. 3a. La posición de cada uno de los 66 puntos de referencia extraídos son usados para construir la máscara, Fig. 3b. La máscara consiste de 89 distancias y éstas forman el vector de distancias, Fig. 3c.

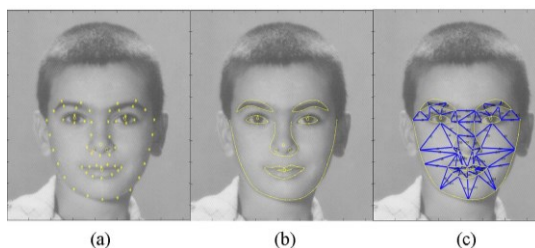


Fig. 3 (a) Los 66 puntos de referencia. (b) Las características faciales. (c) Las 89 distancias entre puntos de referencia

4.1. Individuos del algoritmo evolutivo

Sea M la máscara del rostro. A partir de ésta, se construye un individuo I formado por los puntos de referencia seleccionados de las características seleccionadas de la máscara, y un sub-vector de distancias, formado por las distancias entre los puntos de referencia seleccionados. La Fig. 4 muestra la estructura genética de un individuo I . Un individuo I ésta compuesto de dos partes principales:

1. PuntosDeReferencia: un vector de longitud n fija, formado por las coordenadas $X - Y$ de los n puntos de referencia seleccionados.

2. SubVectorDistancias: un vector de longitud m fija, con las distancias entre cada uno de los puntos de referencia seleccionados.

El algoritmo genera diferentes individuos, por ejemplo, si la característica seleccionada fuese la boca, la Fig. 5 muestra un conjunto de individuos generados por el Algoritmo Evolutivo, cada uno representa una característica mutada con una calificación diferente dependiendo de las proporciones (distancias entre los puntos de referencia) del individuo I ya colocado en la máscara.

PuntosDeReferencia					
X_1	X_2	X_3	X_4	...	X_n
Y_1	Y_2	Y_3	Y_4	...	Y_n
SubVectorDistancias					
D_1	D_2	D_3	D_4	...	D_m

Fig. 4 Representación genética de un individuo

Recuérdese que un individuo sólo contiene información de los puntos de referencia seleccionados y el sub-vector de distancias, de esta forma el algoritmo evolutivo manipula la máscara completa y una lista de restricciones del individuo.

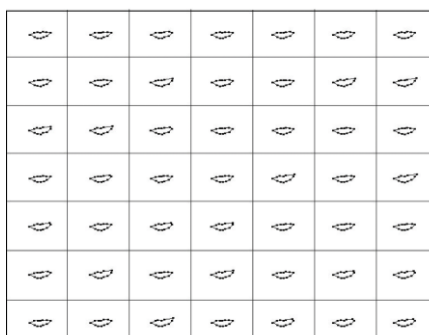


Fig. 5 Bocas -individuos- generadas por evolución usando mutación por escala

4.2. Extracción y unión

Un individuo es extraído directamente de la máscara completa M , mientras esta contenga un conjunto de puntos de referencia seleccionados. A esta operación se le llama Extracción y se usa la notación Extraer (M) para indicar la extracción de un nuevo individuo I de M . Extracción es usada por el Algoritmo evolutivo para producir el primer individuo.

Es posible combinar la máscara completa con un individuo, esta operación es llamada Unión. La notación $\text{Unir}(I, M)$ representa la unión de la máscara completa M con el individuo I . La operación unión actualiza las coordenadas de todos los puntos de referencia seleccionados en M de acuerdo a la información en I . M podría ya tener un diseño para los elementos seleccionados, el cual es reemplazado por la información en el individuo.

Extracción y Unión son usadas por el Algoritmo evolutivo en las diferentes etapas de mejoramiento de la máscara. Por ejemplo, el primer individuo producido es extraído de la máscara de entrada. Posteriormente nuevos individuos son unidos con esta máscara para crear una máscara completa para ser evaluada. Estas operaciones se pueden observar en la Fig. 6.

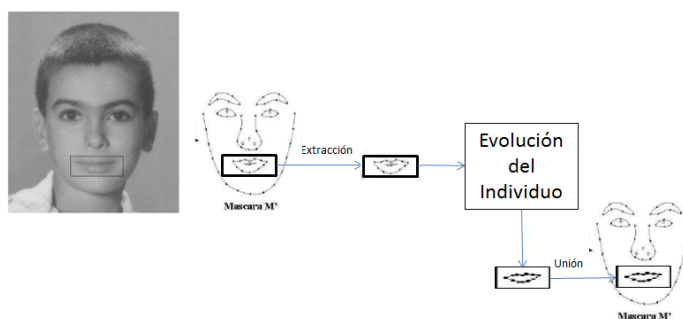


Fig. 6 Operaciones extracción y unión

4.3. Función de evaluación

La función de evaluación proporciona una calificación de belleza b a cada máscara modificada. Un Support Vector Regression (SVR) es construido basado en entrenamiento con un conjunto de ejemplos (x, y) , donde $x \in R^d$ y $y \in R$. El Algoritmo evolutivo utiliza un modelo SVR, entrenado anticipadamente usando la calificación de belleza generado por [6].

El vector de distancias de 89 dimensiones y su correspondiente calificación b de belleza (en una escala de 1 a 7) son usados como ejemplo de entrenamiento para construir un modelo SVR. El SVR define una función $f_b: R^d \rightarrow R$, la cual se usa para estimar las calificaciones de belleza, de los vectores de distancias de los rostros que no se encuentran el conjunto de entrenamiento.

Sea v el vector de distancias normalizado extraído de las características seleccionadas en la máscara de una imagen facial de entrada. El objetivo del proceso de embellecimiento usando Algoritmos evolutivos es crear un vector vecino v' con una calificación de belleza más alta $f_b(v') > f_b(v)$.

El modelo SVR verifica cada una de las proporciones (distancias) de la máscara del rostro formado al unir la máscara M' (una copia de M) y un individuo I , para proporcionar una calificación de belleza, como se muestra en la Fig. 7. En general, las

calificaciones más altas tienen como resultado más belleza y por consiguiente son una mejor solución.

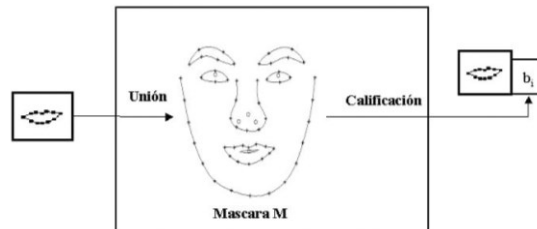


Fig. 7 Función de Evaluación

A cada individuo producido por el sistema es asignando una calificación. Sean dos individuos I_1 e I_2 con una calificación de belleza b_1 y b_2 respectivamente, se dice que I_1 es mejor que I_2 , denotado por $I_1 > I_2$, si b_1 es mas grande que b_2 . Usando este ordenamiento, el objetivo es producir mascarar con una calificación mayor que la máscara de la imagen de entrada, lo que nos lleva al Ciclo Evolutivo.

4.4. Ciclo evolutivo

La entrada del algoritmo evolutivo es una máscara M inicial obtenida del rostro de entrada y una lista R de restricciones. El algoritmo produce una población de individuos representada por sub-mascaras diferentes a la máscara de entrada.

Entonces, el algoritmo evolutivo ejecuta los siguientes pasos:

1. Crea una copia M' de la máscara M . La copia es usada durante todo el proceso del algoritmo.
2. Una población inicial P_0 es creada extrayendo un individuo I de M' y aplicándole los operadores de mutación reiteradamente a I , para producir nuevos individuos.
Se usa una población de 21 individuos. P_0 contendrá el primer individuo I y 20 versiones mutadas.
3. Se inicia un contador de generaciones $iter = 0$.
4. P_{iter} es evaluada para calcular una calificación b_i para cada individuo en la población.
5. Una Sub-población S es seleccionada de P_{iter} por *certamen*: el mejor individuo en P_{iter} es directamente insertado en S (El mejor individuo es aquel que tiene la mejor calificación proporcionada por la función de evaluación). El resto de individuos en P_{iter} toman parte en competencias partwise: para cualquier I_1 e I_2 que aún no han participado en una competencia y elegidos aleatoriamente de P_{iter} , si $I_1 < I_2$ se agrega I_1 en S ; en caso contrario se agrega en I_2 a S . En total 10 concursos partwise son ejecutados, resultando 10 ganadores. Al final de este paso, S tiene 11 individuos.
6. $iter$ es incrementado $iter \leftarrow iter + 1$

7. Una nueva población, P_{iter} , es creada: el mejor individuo en S (decidido en el paso 5) es transferido directamente a P_{iter} ; entonces dos individuos I_1 e I_2 son elegidos aleatoriamente y removidos de S . Estos individuos son combinados por un operador de cruce y son agregados a P_{iter} . El proceso se repite hasta que S este vacía. Para completar la nueva población, los otros 10 individuos son producidos por mutación del mejor individuo en P_{iter-1} .
8. Si se alcanza el criterio de paro, entonces termina el ciclo evolutivo, y la salida del algoritmo evolutivo es P_{iter} ; en otro caso, se regresa al paso

En el algoritmo evolutivo, los nuevos individuos son creados a través de mutación o combinación (cruza) de los existentes. Buenos individuos (con las calificaciones más altas), se espera que tengan éxito en el torneo selectivo y propaguen sus características a las siguientes generaciones. El algoritmo evolutivo se detiene hasta que la máscara del mejor individuo ya no tiene mejoras después de L iteraciones consecutivas del ciclo, y sólo $50 * L$ generaciones tienen que ser producidas. La Fig. 8 ilustra la funcionalidad interna del algoritmo evolutivo.

4.5. Operadores

En esta subsección se describen los operadores de mutación y cruce implementados en el algoritmo evolutivo propuesto. La cruce combina información genética y requiere de dos individuos, mientras que la mutación cambia al individuo mismo.

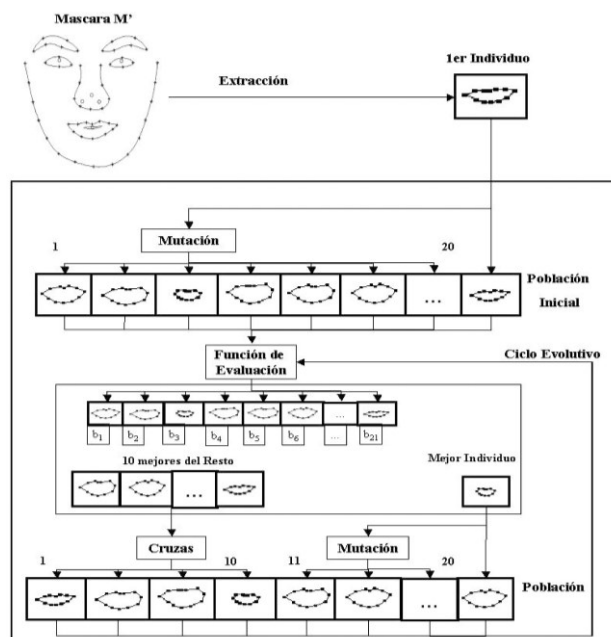


Fig. 8 Funcionalidad Interna del Algoritmo evolutivo

Mutación. La cruce sólo puede encontrar nuevas combinaciones de material genético ya presente. Para introducir nuevo material genético en la población, el cambio aleatorio de individuos es necesario por ejemplo a través de la mutación. En nuestra propuesta se implementaron cinco operadores de mutación.

ESCALAR (Individuo I). Este operador cambia las coordenadas de los puntos de referencia seleccionados que conforman el individuo I . Aleatoriamente se elige un número n de iteraciones, después por cada iteración: se elige aleatoriamente un punto de referencia a , una dirección de escala $d = -1$ (Reducir) o $d = 1$ (Aumentar) y una escala S dependiendo de la dirección, si $d = -1$ entre $[-1, 0)$ y si $d = 1$ entre $(0, 1]$ y se toman en cuenta las restricciones que el usuario introdujo, si la escala rebaza un límite. La estrategia para escalar un objeto con respecto al punto arbitrario P_1 es la siguiente: primero se hace el traslado de manera que P_1 pase al origen, después se escala y luego se traslada de regreso a P_1 . La Ecuación (1) muestra la matriz resultante de multiplicar las tres transformaciones mencionadas anteriormente, necesarias para escalar. El punto arbitrario P_1 representado en la ecuación con su coordenada (x_1, y_1) es el punto medio del individuo. $S(sx, sy)$ es la escala en x y y respectivamente.

$$T(x_1, y_1) * S(sx, sy) * T(-x_1, -y_1) = \begin{pmatrix} sx & 0 & x_1(1-sx) \\ 0 & sy & y_1(1-sy) \\ 0 & 0 & 1 \end{pmatrix} \quad (1)$$

ROTAR (Individuo I). Este operador cambia las coordenadas de los puntos de referencia seleccionados del individuo I . Aleatoriamente se elige un número n de iteraciones, después por cada iteración: se elige aleatoriamente un punto de referencia a , un grado a rotar α (entre 0° y 45°) y una dirección de rotación $d = -1$ (Derecha) o $d = 1$ (Izquierda), si la rotación sobre pasa los límites de las restricciones, se ejecuta otra iteración.

En este caso, la transformación de rotar utiliza la Ecuación 2. Esta mutación utiliza una estrategia similar al de *ESCALAR*. La rotación es con respecto a un punto arbitrario P_1 : primero se traslada P_1 al origen, se rota y se traslada para que el punto en el origen regrese a P_1 . El punto arbitrario representado con la coordenada (x_1, y_1) en la ecuación, es el punto medio del individuo.

$$T(x_1, y_1) * R(\theta) * T(-x_1, -y_1) = \begin{pmatrix} \cos\theta & -\text{sen}\theta & x_1(1-\cos\theta)+y_1\text{sen}\theta \\ \text{sen}\theta & \cos\theta & y_1(1-\cos\theta)+x_1\text{sen}\theta \\ 0 & 0 & 1 \end{pmatrix} \quad (2)$$

TRASLADAR (Individuo I). Este operador cambia las coordenadas de los puntos de referencia (seleccionados), que conforman el individuo I . Aleatoriamente se elige un número n de iteraciones, después por cada iteración: se elige aleatoriamente un punto de referencia a , una dirección (arriba-abajo o vertical-horizontal) y un desplazamiento d .

El valor a d es de acuerdo a una distribución lineal de probabilidades donde los valores pequeños de desplazamiento tiene una probabilidad más alta de ser elegidos, si el desplazamiento rebaza un límite, esa iteración no es tomada en cuenta. La traslación se realiza utilizando la Ecuación 3, donde el desplazamiento en x y y es representado por (dx, dy) respectivamente

$$T(dx,dy)= \begin{pmatrix} 1 & 0 & dx \\ 0 & 1 & dy \\ 0 & 0 & 1 \end{pmatrix} \quad (3)$$

CAMBIO-ALEATORIO (Individuo I). Este operador cambia las coordenadas de un punto de referencia a en el individuo I . Aleatoriamente se elige un punto de referencia a , una dirección (arriba-abajo o vertical-horizontal) a mover y un desplazamiento $k \in Z^*$. El valor a k es de acuerdo a una distribución lineal de probabilidades donde los valores pequeños absolutos de desplazamiento tienen una probabilidad más alta de ser elegidos, tomando en cuenta las restricciones que el usuario introdujo.

CAMINAR (Individuo I). Este operador sigue la misma idea de la mutación Walk-BC introducida por [7]. Nuestra versión de Walk-BC solo mueve los puntos de referencia seleccionados. Aleatoriamente se elige un punto de referencia a del individuo I y una dirección vertical $d = 1$ (para Arriba) o $d = -1$ (para Abajo). La coordenada X de a en I es igual a la media aritmética de todos los puntos de referencia adyacentes de a en la capa $v.l - d$. Entonces un punto de referencia u adyacente a v en la dirección d es elegido aleatoriamente y el proceso se repite para u . La mutación continúa hasta que ya no haya más puntos de referencia adyacentes a elegir en la dirección d .

Cruza. El propósito del operador de cruza es crear nuevos individuos que se espera que sean mejores que los padres. El algoritmo evolutivo tiene dos operadores de cruza:

COMBINAR-PUNTOS (Individuo I_1, I_2). Elige aleatoriamente n puntos de referencia a_n que tiene diferentes coordenadas en I_1 e I_2 , y cambia las posiciones $X - Y$ de todos los n puntos de referencia entre los individuos.

COMBINAR-PARTES (Individuo I_1, I_2). Elige aleatoriamente un punto de referencia a , los nuevos individuos están conformados como sigue: I_{1Hijo} está formado por los puntos de referencia anteriores a a de I_1 y los puntos de referencia a partir de a hasta el final de I_2 , y I_{2Hijo} está formado por los puntos de referencia anteriores a a de I_2 y los puntos de referencia a partir de a hasta el final de I_1 .

5 Resultados

En la Fig. 9 se muestran resultados del embellecimiento evolutivo, los resultados son evidentes, la posibilidad de la obtención de “muchas versiones del YO original”, algunas embellecidas más que otras.



Fig. 9. Las imágenes al centro y der. son versiones embellecidas que se obtuvieron evolucionando el original a la izq. La imagen de la der. tiene mayor puntuación de belleza que la del centro. Imagen izq. (imagen original) tomada de Huiyun, et al. [8]

La Fig. 10 y Fig. 11 muestran más resultados que se obtuvieron utilizando el sistema implementado en este artículo. Es importante mencionar que nuestro algoritmo evolutivo esta implementado en Matlab y el tiempo de ejecución fue de aproximadamente 45 segundos.

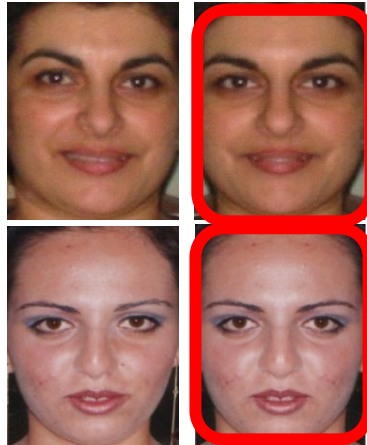


Fig. 10. Algunos resultados haciendo uso del algoritmo evolutivo. Las imágenes de la der. tienen mayor puntuación de belleza que las de la izquierda. Nótese cómo la boca a sufrido cambios positivos en ambos rostros.

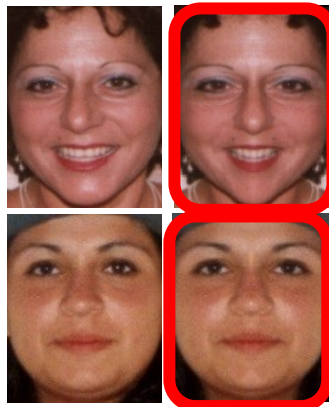


Fig. 11. Otros resultados haciendo uso del algoritmo evolutivo. Las imágenes de la der. tiene mayor puntuación de belleza que las de la izquierda. Nótese cómo la boca ha sufrido cambios positivos en ambos rostros. La imagen superior también muestra cambios en el mentón.

Actualmente nuestra técnica está limitada a rostros de vista frontal y bajo una expresión neutral, en los resultados experimentales se utilizó base de datos de imágenes Fg-Net [9].

6. Conclusiones

La innovación de este trabajo son las características de *ENFOCAR*, para embellecer algunas partes del rostro de entrada. En nuestra propuesta se le permite al usuario sólo trabajar con una parte del rostro a través de la máscara, mejorando únicamente la característica seleccionada que no le son favorables en la imagen. Se diseñó un algoritmo evolutivo para generar un rostro más bello que el de entrada.

Referencias

1. Moshe, B., Maital, N. and Heather, L.: Very first impressions. *Emotion by the American Psychological Association*, 6(2):269–278 (2006)
2. Winkielman, P., Halberstadt, J., Fazendeiro, T. and Steve, C.: Prototypes are attractive because they are easy on the mind. *Psychological Science*, 17(9):799–806 (2006)
3. Pawan, S., Benjamin, B., Ostrovsky, Y. and Richard, R.: Face recognition by humans: 20 results all computer vision researchers should know about. Technical report, Department of Brain and Cognitive Sciences, Massachusetts Institute of Technology (2005)
4. Eysenck, H., Dror, G. and Ruppin, E.: Facial attractiveness: Beauty and the machine. *Neural Computation*, 18(1):119–142 (2006)
5. Leyvand, T., Cohen Or, D., Dror, G. and Lischinski, D.: Data Driven Enhancement of Facial Attractiveness. *ACM Trans. Graph.*, 27(3):9 (2008)
6. Kagian, A., Dror, G., Leyvand, T., Cohen-Or, D. and Ruppin, E.: A Humanlike Predictor of Facial Attractiveness. In: Schölkopf B., Platt J., and Hoffman T., editors, *Advances in Neural Information Processing Systems*, volume 19. MIT Press. (2007)
7. Utech, J., Branke, J., Schmeck, H. and Eades, P.: An Evolutionary Algorithm for Drawing Directed Graphs. In: *International Conference on Imaging Science, Systems, & Technology (CISST'98)*, pages 154–160 (1998)
8. Huiyun, M., Lianwen, J. and Minghui, D.: Automatic Classification of Chinese Female Facial Beauty using Support Vector Machine. In: *Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics*, pages 4987– 4991 (2009)
9. Cootes, T.: The Fg-Net Aging Database, <http://sting.cycollege.ac.cy/~alanitis/fgnetaging/index.htm>

Eigenespacios de belleza paramétricos como máquina calificadora

Ricardo Solano Monje¹, Nayeli Joaquinita Meléndez Acosta²,
Sergio Juárez Vázquez¹, Homero Vladimir Ríos Figueroa³

Universidad del Istmo, Campus Sto. Domingo Tehuantepec, Oax.,
México

² Universidad del Istmo, Campus Ixtepec, Oax.,
México

³ Universidad Veracruzana, Xalapa, Ver.,
México

rsolanomonje@gmail.com, sjuarez@bianni.unistmo.edu.mx,
nayelimelendez@gmail.com, hrios@uv.com.mx

Resumen. Este artículo presenta una máquina calificadora de belleza facial. El concepto de Máquina calificadora se debe a Eisenthal et al [4]. La máquina calificadora es implementada haciendo uso de un Eigenespacio de belleza paramétrico, tomando como ensamble vectores de distancias entre puntos de referencia de rostros. La máquina calificadora propuesta parte de la selección asistida de un conjunto de características faciales predefinidas, a partir de estas características faciales y de la posición de los puntos de referencia relativos a la imagen facial, se calcula un conjunto de distancias entre dichas características, se localizan 66 puntos de referencia, con estos puntos se construye una malla o máscara. El objetivo final de la máscara es formar un vector de distancias o longitudes entre los puntos de referencia que definen las características faciales. Estos vectores de distancias son utilizados para contruir un Eigenespacio. Las curva paramétrica acarrear la calificación de belleza de cada rostro del ensamble. La forma más simple de determinar la calificación de un rostro es proyectarla al Eigenespacio y buscar el vector más cercano. Mostramos cómo esto es suficiente para dar una calificación acertada a un rostro de entrada original.

Palabras clave: máquina calificadora, belleza facial, eigenespacios.

1. Introducción

La belleza ha captado la atención de los seres humanos desde que inicio la humanidad. La gente usa diferentes formas para que sus rostros se vean más atractivos, tales como maquillaje y cirugía cosmético. Para ejemplo, utilizamos bases para cambiar el tono de la piel, usar corrector para arrugas y pecas [1].

Esta noción del atractivo facial junto con la capacidad para evaluar de forma fiable y automáticamente la belleza facial de una imagen ha motivado este trabajo.

El componente clave de nuestro enfoque es un motor calificador de la belleza entrenado a partir de un conjunto de imágenes de rostros con el acompañamiento de una calificación de belleza proporcionado por un conjunto de evaluadores humanos.

El contenido de este artículo está organizado en cinco secciones de la siguiente manera: la segunda sección trata las bases teóricas de la aproximación de Eigenespacios. En la tercera sección se explica el proceso de embellecimiento propuesto por Leyvand et al [3], particularmente la extracción de los puntos de referencia que serán los parámetros de belleza de la máquina calificadora. La cuarta sección explica la propuesta de este artículo. Finalmente en la quinta sección se muestran las conclusiones y trabajo futuro.

2. La aproximación de eigenespacios

El reconocimiento de objetos comprende al menos tres etapas, la primera la creación del Eigenespacio a partir del ensamble de imágenes disponibles, la segunda la representación de las imágenes del ensamble en el Eigenespacio y por último la clasificación de imágenes originales, previa representación en el Eigenespacio.

2.1. La transformada K-L y reducción de dimensionalidad

El término Eigenespacio Universal se utiliza en la aproximación de apariencia para denotar el modelo que representa a un conjunto fijo de objetos. Tal modelo se genera a partir del ensamble de imágenes disponibles de los objetos a reconocer. Dicho modelo es representado por un subespacio particular llamado Eigenespacio. Para obtener el Eigenespacio a partir del ensamble, se aplica la transformada Karhunen-Loève abreviada K-L. La transformada K-L es un método bien estudiado para la compresión de información. En el caso discreto, donde la información es representada por un conjunto de vectores-imagen llamado ensamble, la transformada Karhunen-Loève minimiza el error cuadrático medio inducido al tomar solo unos términos de la expansión del total de vectores ortonormales necesarios para reconstruir el conjunto de vectores del ensamble. El conjunto de vectores ortogonales que se obtienen al aplicar la transformada K-L al ensamble es frecuentemente llamado componentes principales y la aplicación de la transformada K-L, análisis de componentes principales. Estos vectores capturan más varianza estadística que cualquier otra base. En la fase de creación del modelo –Eigenespacio- se obtienen M imágenes de $S \times R = N$ pixeles de resolución que forman el ensamble B

$$B = \{I_1, I_2, \dots, I_M\}$$

Típicamente la imagen media \bar{x} del ensamble

$$\bar{x} = \frac{1}{M} \sum_{i=1}^M I_i \quad (1)$$

es sustraída de cada imagen del ensamble resultando en el conjunto

$$X = \{I_1 - \bar{x}, I_2 - \bar{x}, \dots, I_M - \bar{x}\}$$

El conjunto X puede ser visto como matriz de $N \times M$. Esto nos permite crear la matriz de covarianza $\Omega = XX^T$ que caracteriza la distribución de las M imágenes en \mathfrak{R}^N . Las componentes principales del ensamble se obtienen descomponiendo la matriz de covarianza Ω en vectores y valores singulares ¹.

Sean \bar{u}_i y λ_i los eigenvectores y eigenvalores de Ω respectivamente, entonces el Eigenespacio o las componentes principales viene dado por

$$U = [\bar{u}_1, \bar{u}_2, \dots, \bar{u}_M]$$

U se utiliza como matriz de transformación de espacios, del espacio de la imagen al espacio de baja dimensión.

Una vez que la base ha sido creada cada uno de los vectores-imagen $x_i \in X$ se proyectan al subespacio de menor dimensión generado por U , por medio de

$$y_i = U^T x_i \quad i = 1 \dots M$$

La forma más simple de determinar la clase de una imagen original es proyectarla al Eigenespacio y buscar el vector más cercano.

3. Proceso de embellecimiento propuesto por Leyvand

Partiendo de la imagen facial de entrada, se determinan una variedad de características faciales predefinidas. Las características faciales -8 en total- están formadas por puntos de referencia y corresponden a 2 cejas, 2 ojos, labio superior, labio inferior, contorno de la nariz y contorno del rostro. A partir de las características faciales y de la posición de los puntos de referencia se calcula un conjunto de distancias entre dichas características. Leyvand et al en [3]. localiza 84 puntos de referencia, mientras que Eisenthal et al. en [4] localiza 37 puntos de referencia. Con estos puntos se construye una malla o máscara. El objetivo final de la máscara es formar un vector de distancias o longitudes entre los puntos de referencia que definen las características faciales.

El trabajo de Leyvand et al. obtiene 234 distancias que se almacenan en un vector. El hecho de que el grupo de Eisenthal et al. localice 37 puntos de referencia y el

¹ Obsérvese que Ω está en $\mathfrak{R}^{N \times N}$. En [2] Turk et al proponen una forma eficiente de obtener los eigenvectores de Ω a partir de $\Omega' = X^T X$. En este caso Ω' está en $\mathfrak{R}^{M \times M}$.

grupo de Leyvand et al. localice 84 tiene una explicación sencilla, de hecho, Leyvand et al. usan la máquina calificadora de Eisenthal et al., pero mientras que Eisenthal et al. persiguen en su trabajo calificar la belleza de un rostro, Leyvand et al. realiza la modificación y embellecimiento de un rostro. Ya que la modificación de la imagen de entrada se llevará a cabo con transformaciones *warping*, se requieren más puntos para realizar una transición suave hacia la imagen de salida, por ello el uso de más puntos de referencia y la construcción de la malla respectiva.

El objetivo de la máscara es la creación de un vector de distancias. Cada vector es de dimensión 234, y para hacerlo invariante a escala, se normaliza con la raíz cuadrada del área del rostro. Sin embargo, es claro que este tipo de normalización puede cambiarse según nuestros intereses, por ejemplo, podría normalizarse dividiendo cada entrada del vector entre la distancia mayor de dicho vector.

Estas distancias que se obtienen de cada rostro definen un punto o vector en un espacio multidimensional -un espacio de rostros particular-. Debe observarse que este espacio de rostros es muy simple pues es meramente un conjunto de vectores con cierto orden de prioridad. Es en este espacio donde se busca el punto más cercano al rostro de entrada, que corresponda al rostro más atractivo del conjunto de rostros atractivos. Esta búsqueda es ponderada por belleza pero aun así el esquema es demasiado simple.

El espacio de rostros está formado por los vectores de distancias -obtenidos de las máscaras- de un conjunto de 92 rostros bellos, previamente seleccionados. Estos 92 rostros bellos fueron calificados por jueces externos al sistema. Las mallas de los 92 rostros junto con su calificación forman el espacio de rostros particular que se define en [5]. En este sentido, y como puede apreciarse, no estamos frente a un espacio de rostros sofisticado.

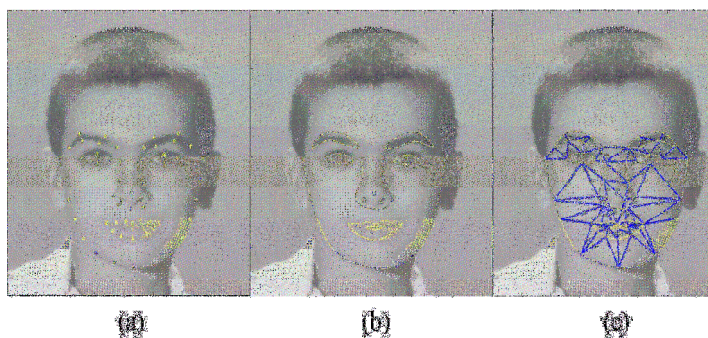


Fig. 1. (a) Los 66 puntos de referencia. (b) Las características faciales. (c) Las 89 distancias entre puntos de referencia

El componente clave del sistema de embellecimiento durante la búsqueda en el espacio de rostros bellos, es la máquina de aprendizaje que califica de forma automática la belleza facial. La máquina calificadora de Leyvand utiliza la técnica SVM Support Vector Machine.

4. Eigespacios de belleza paramétricos

La propuesta de este artículo plantea el uso de Eigespacios como los desarrollados por Nayar et al. en [6] para hacer una función calificadora de belleza con Eigespacios, tomando como ensamble los vectores de puntos de referencia de rostros bellos y feos.

En nuestro caso y para fines explicativos, se considera que partiendo del rostro de entrada se extrae un total de 66 puntos de referencia de manera manual como se muestra en la Fig. 1. Al igual que en [3], los puntos de referencia son localizados en los contornos de las 8 diferentes características faciales: las dos cejas, los dos ojos, el labio inferior y superior de la boca, la nariz y límites del rostro. La posición de cada uno de los 66 puntos de referencia extraídos son usados para construir la máscara. La máscara consiste de 89 distancias y éstas forman el vector de distancias.

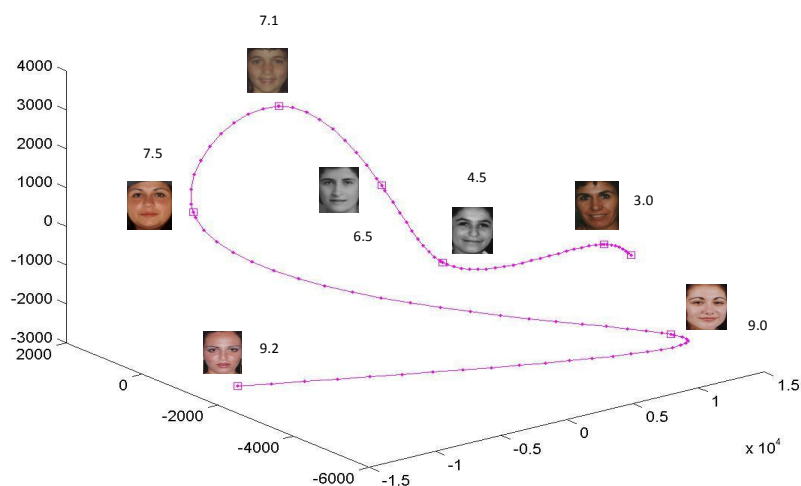


Fig. 2. Curva paramétrica en el Eigespacio de máscaras de los rostros del ensamble. Los vectores-máscaras se interpolan, permitiendo localizar con mayor precisión la mejor calificación. Cada vector del ensamble original así como todos los puntos interpolados tienen asociado o acarrean una calificación de belleza, que permite calificar un rostro de entrada original.

Usando los vectores obtenidos de las máscaras de cada imagen del ensamble se crea un Eigespacio y se mapean cada una de las máscaras, según se explicó la creación del Eigespacio y el mapeo en la sección 2. Una vez hecho el mapeo de cada máscara al Eigespacio, se interpolan los vectores mapeados y se crea una curva paramétrica entre cada uno de estos vectores que representan a las máscaras de los rostros del ensamble. La curva paramétrica acarrean la calificación de belleza. Para calificar una imagen rostro de entrada original, se obtiene la máscara de la imagen, se mapea al Eigespacio y usando la curva paramétrica interpolada se encuentra el vector que más se le aproxima al mapeado, finalmente dicho vector similar

proporciona la calificación que acarrea consigo que le fue previamente asignada. Una representación 3D de la curva paramétrica se observa en la Fig. 2. Vale mencionar que la curva en el Eigenespacio es de dimensión 66 o menor, la mostrada en la Fig. 2 es de dimensión 3 para propósitos ilustrativos.

Una vez creada la curva paramétrica entre cada uno de los vectores que representan a las máscaras de los rostros del ensamble y ya que la curva paramétrica acarrea la calificación de belleza es necesario realizar una segunda interpolación lineal para proporcionar una calificación a los vectores de las máscaras interpolados.

Teniendo la curva paramétrica la de las máscaras de rostros y la interpolación lineal de sus calificaciones, dada una imagen de entrada se busca la máscara del rostro que más se le aproxime, localizada esta máscara obtenemos su calificación y esta calificación es proporcionada al rostro de entrada.

5. Resultados

Nuestros resultados están basados en un ensamble de 50 de imágenes de rostros de mujeres con su calificación de belleza respectiva. En la Fig. 3 se muestran algunas de las imágenes usadas en el ensamble. Todo el desarrollo está basado en la base de imágenes Fg-Net [10].

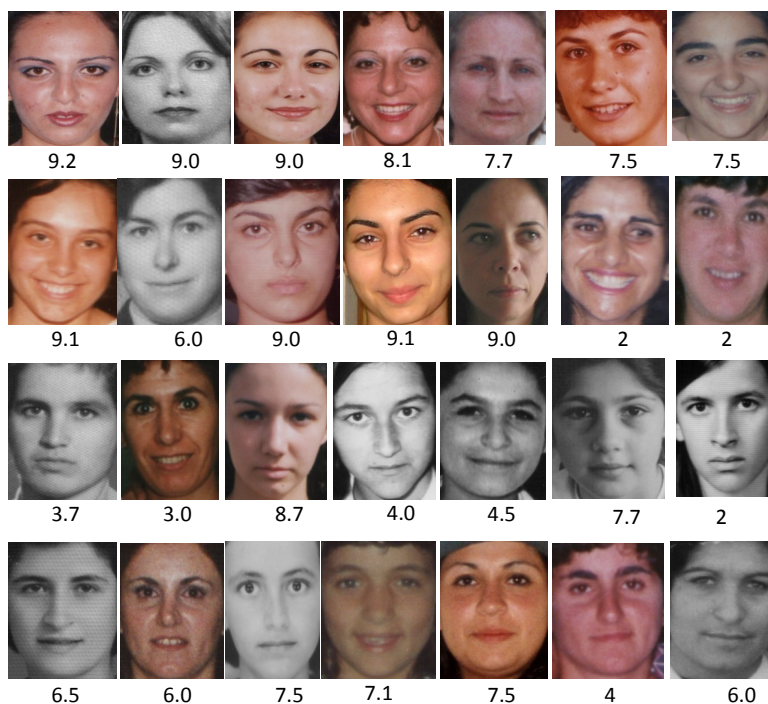


Fig. 3. 28 imágenes usadas en el ensamble

El espacio de rostros está formado por los vectores de distancias -obtenidos de las máscaras- de un conjunto de 50 imágenes de rostros diferentes, previamente previo marcado de la máscara.

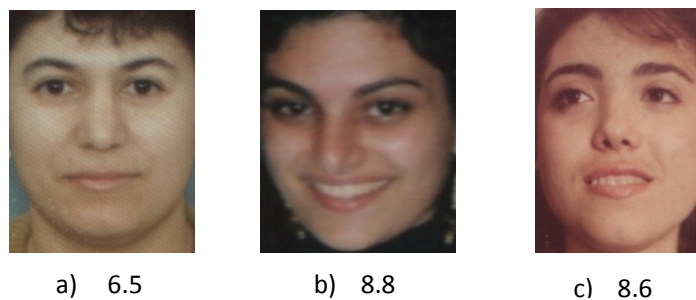


Fig. 4. Algunos resultados obtenidos de utilizar el método propuesto en este artículo.

La Fig. 5 muestra los resultados obtenidos sobre la curva paramétrica, podemos observar que la imagen a) de la Fig. 4 es la única que es equivalente a una máscara de los rostros de ensamble, las otras son semejantes a máscaras creadas por interpolación.

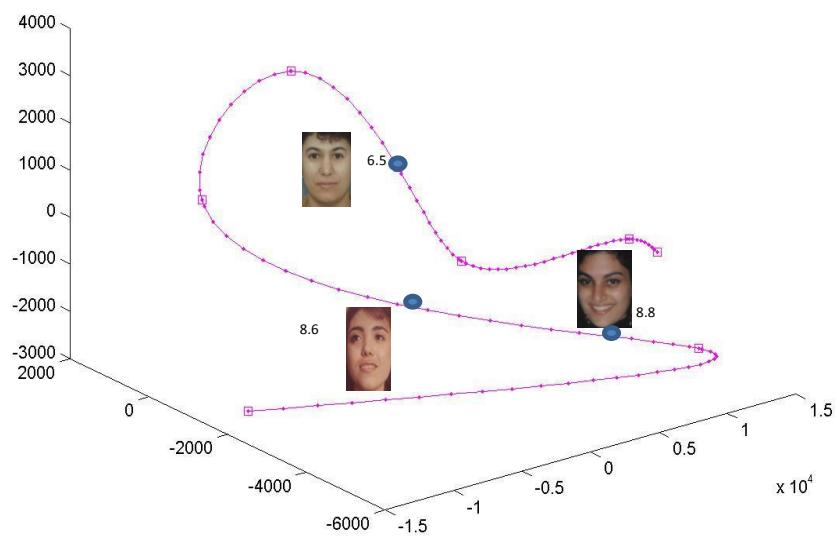


Fig. 5. Resultados sobre la curva paramétrica.

6. Conclusiones y trabajo futuro

Hemos mostrado como nuestra propuesta es factible. Nuestra aproximación tiene aspectos que son deseables en aplicaciones de respuesta casi inmediata. Los submodelos pueden agruparse en clusters según vayan siendo construidos o conforme las imágenes de los rostros vayan siendo adquiridas. Es evidente que en nuestra aproximación se pueden agregar nuevos rostros o eliminar existentes en un tiempo razonable.

La función de evaluación de belleza puede realizarse también con una red neuronal. De hecho cualquier aproximación de aprendizaje supervisado permite aproximar dicha función.

Referencias

1. Zhao, Y., Huang, X., Gao, J., Tokuta, A., Zhang, A., Yang, R.: Video Face Beatification (2013)
2. Turk, M., Pentland, A.: Face recognition using eigenfaces, Proc. Of IEEE Conf. On Comp. Vision and Pattern Recog., 586–591 (1991)
3. Leyvand, T., Cohen Or, D., Dror, G., Lischinski, D.: Data Driven Enhancement of Facial Attractiveness. ACM Trans. Graph., 27(3):9 (2008)
4. Eishenthal, Y., Dror, G., Ruppin, E.: Facial attractiveness: Beauty and the machine. Neural Computation, 18(1):119–142 (2006)
5. Bar, M., Neta, M., Linz, H.: Very first impressions. Emotion by the American Psychological Association, 6(2):269–278 (2006)
6. Nene, S., Nayar, S., Murase, H.: Software library for appearance matching. Technical Report CUCS-019-94, Department of Computer Science, Columbia University (1994)
7. Sirovich, L., Kirby, M.: Application of the Karhunen-Loeve procedure for the characterization of human faces. IEEE Transactions on Pattern Analysis and Machine Intelligence, 12(1) (1990)
8. Altamirano, L.C., Altamirano, L., Alvarado, M.: Non-Uniform sampling for Improve Appearance-based Models. Pattern Recognition Letters (2002)
9. Chandrasekaran, S., Manjunath, B., Wang, Y., Zhang, H.: An Eigenspace Update Algorithm for Image Analysis, Graphical Models and Image Processing, 59(5):321–332 (1997)
10. Murase, H., Nayar, S.K.: Parametric Eigenspace Representation for Visual Learning and Recognition. In: Proceedings of The International Society for Optical Engineering (SPIE), volume 2031, pp. 378–391 (1993)
11. Nayar, S.K., Murase, H., Nene, S.A.: Parametric Appearance Representation. In: Early Visual Learning, pp. 131–160 (1996)
12. Nayar, S.K., Baker, S., Murase, H.: Parametric Feature Detection. In: DARPA Image Understanding Workshop (IUW), pp. 1425–1430 (1997)
13. Cootes, T.: The Fg-Net Aging Database, <http://sting.cycollege.ac.cy/~alanitis/fgnetaging/index.htm>

An Estimation of Distribution Algorithm based Approach for the Order Batching Problem

Ricardo Pérez-Rodríguez, Arturo Hernández-Aguirre

CIMAT A.C., Center for Research in Mathematics,
Guanajuato, Guanajuato, Mexico

ricardo.perez@cimat.mx, artha@cimat.mx

Abstract. Up to 50 per cent of all warehouse operations costs can be attributed to order picking [1]. It is an aspect critical to each supply chain and the planning and control of warehouse processes. According to experience from logistics environments, a reduction of picking times is an integral part of the delivery lead-time and may result in an improvement of the customer service provided by a warehouse. Order batching consists of combining customer orders into picking orders to minimize the picking time. Different evolutionary algorithms have been proposed to build better batches for the order picking. These algorithms try to preserve batches or sequences of batches that appear frequently in all members of the population in order to keep track and inherit these characteristics exhibited by the parents to the next generation. However, the traditional evolutionary operators used in current research may lose the characteristics mentioned. To address this situation, an estimation of distribution algorithm - based approach is proposed for the order batching problem in an order-picking warehouse. A probability model is built to describe the characteristics exhibited by the parents as a distribution of the solution space. In a series of extensive numerical experiments, an acceptable performance using the algorithm mentioned is shown against different evolutionary algorithms known in the literature.

Keywords: estimation of distribution algorithm, warehouse management, order picking, order batching.

1 Introduction

When different incoming articles are received and stored in unit loads such as pallets or racks, and the customers require a few quantities of different articles, the order picking arises [2]. Order picking systems involving operators are very common in industry. Such manual order picking systems can be divided into two categories [3]: parts-to-picker, where automated storage and retrieval systems deliver the articles to a transfer site where stationary pickers are located; picker-to-parts, where pickers normally walk or ride (e.g. on an automated guided vehicle) through the warehouse and collect articles. If the number of arriving orders is too large for processing each cus-

customer order separately in an appropriate amount of time, customer orders must be combined into batches. The process of grouping a set of customer orders into picking orders is referred to as order batching. It means that different customer orders can be simultaneously released for picking. Manual picking systems such as picker-to-parts where the activity of transformation of customer orders into batches (picking orders) is analyzed in this paper.

On the tours through the warehouse, order pickers are guided by pick lists. A pick list comprises a set of order lines, each one identifying a particular article, the quantity of the article requested and the respective storage location. The order lines are already sorted into the sequence according to which the order picker is meant to collect the items.

The order batching problem is a combinatorial issue. It features some similarities with the capacitated vehicle routing problem, however differs from that with respect to the customer order integrity condition, i.e. all items of a customer order must be picked on the same tour. Thus, traditional solutions approaches to the capacitated vehicle routing problem cannot be applied directly to the order batching problem [4].

The order batching has been studied and solved through different evolutionary algorithms such as Genetic Algorithms (GA), Memetic Algorithms (MA), and Tabu Search (TS). A main characteristic of these algorithms is to preserve batches or sequences of batches that appear frequently in all members of the population in order to keep track and inherit these characteristics exhibited by the parents to the next generation. However, the traditional evolutionary operators used in current research may lose the characteristics mentioned. An example can be found in [5].

Another approach consists of using Estimation of Distribution Algorithms (EDA), introduced by [6]. EDA is a relatively new paradigm in the field of evolutionary computation. Compared with other evolutionary algorithms, the EDA reproduces new population implicitly instead of using traditional evolutionary operators. In the EDA, a probability model of the most promising area is built by statistical information based on the search experience to describe the characteristics exhibited by the parents and then the probability model is used for sampling to generate new individuals. The EDA makes use of the probability model to describe the distribution of the solution space. The updating process reflects the evolutionary trend of the population. To the best of our knowledge, this kind of algorithm has not been used to tackle the order batching problem in order-picking warehouses.

2 Problem Statement

Based on [7], the order batching problem and an optimization model are explained below.

The picking process being analyzed here is described in the following way: the operators (pickers) start at a depot, the picker walks through the warehouse and picks up articles from different storage locations. Afterwards, the picker returns to the depot and hands over the picked articles. The route considered by the picker in order to find and collect articles in this research is the “S-Shape” route, because it is easy to under-

stand and has been widely used in industrial environments. Figure 1 shows an example of the “S-Shape” route for a set of articles to be picked. The black rectangles represent pick locations, i.e., the corresponding locations where articles have to be picked.

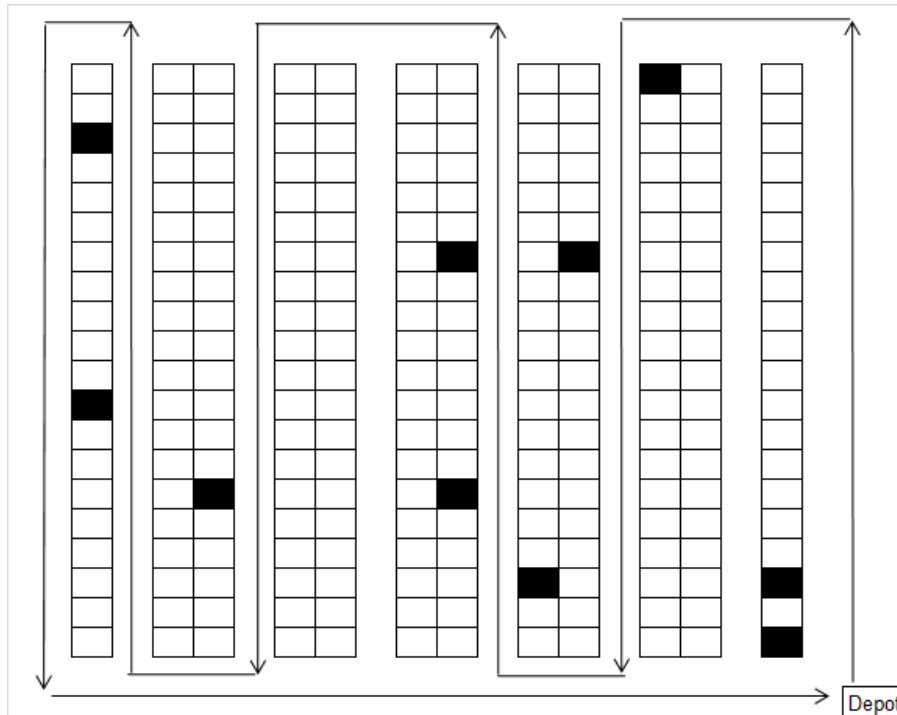


Fig. 1. Example of “S-Shape” route.

The order picking process is usually done with the help of a picking device. Consequently, different orders can be combined until the capacity of the device is exhausted. The splitting of an order into two or more batches is prohibited, since it would result in additional unacceptable sorting efforts. If the picker has already started a tour, it is completed without interruption.

In this research a single order picker is considered, i.e., all batches must be processed one after another. We focus on minimizing the total completion time of all customer orders. Therefore, when a set of unprocessed orders arrives and an order picker becomes available, those unprocessed orders can be grouped into one or more batches that should be released directly, or its start should be postponed to a later point in time.

Let

v_{travel} : travel velocity, i.e. number of length units the picker can cover in the warehouse per time unit.

v_{pick} : pick velocity, i.e. number of items the picker can search and pick per unit time.
 n : be the number of customer orders known,
 m : the number of batches to be processed,
 a_i : the arrival time of order i for all $i \in \{1, \dots, n\}$,
 k_i : the number of articles of order i for all $i \in \{1, \dots, n\}$,
 K : the maximal number of articles that can be included in any batch (device capacity),
 S_j : the start time of the batch j for all $j \in \{1, \dots, m\}$
 $x_{ji} = \{1 \text{ if order } i \text{ is assigned to batch } j, 0 \text{ otherwise}\}$
 $\mathbf{x}_j = (x_{j1}, \dots, x_{jn})^T$ is a vector where some customer orders are associated to the batch j , for all $j \in \{1, \dots, m\}$

An optimization model can be formulated as follows:

The total length of the picking tour for a particular batch involving the S-shape routing method is determined by d as a function for determining the length of tour given a batch. The problem can be modeled as follows:

$$\text{Min } \max_{j \in \{1, \dots, m\}} \left\{ S_j + \frac{d(\mathbf{x}_j)}{v_{travel}} + \frac{\sum_{i=1}^n k_i x_{ji}}{v_{pick}} \right\} \quad (1)$$

$$\sum_{j=1}^m x_{ji} = 1, \forall i \in \{1, \dots, n\} \quad (2)$$

$$\sum_{i=1}^n k_i x_{ji} \leq K, \forall j \in \{1, \dots, m\} \quad (3)$$

$$S_j \geq \max_{i \in \{1, \dots, n\}} \{a_i \cdot x_{ji}\}, \forall j \in \{1, \dots, m\} \quad (4)$$

$$S_j \geq S_{j-1} + \frac{d(\mathbf{x}_{j-1})}{v_{travel}} + \frac{\sum_{i=1}^n k_i x_{j-1i}}{v_{pick}}, \forall j \in \{2, \dots, m\} \quad (5)$$

$$S_j \geq 0, \forall j \in \{1, \dots, m\} \quad (6)$$

$$x_{ji} \in \{0, 1\} \quad (7)$$

In the objective function (1) the expression after S_j represents the service time of a batch. This sum is composed of the time the picker needs to travel and the time he/she needs to pick the items. Equations (2) ensure the assignment of each order to exactly one batch. Inequalities (3) guarantee that the capacity of the picking device is not violated. The conditions (4) indicate that a batch is started after all customer orders assigned to that batch are known. From (5) follows that a batch is started after the previous one is completed. Finally (6) and (7) indicate that the start times are non-negative and the variables are binary.

3 Literature Review

A discussion about the most current research on the order batching process is outlined below.

[8] analyzed a practical batching problem where greeting cards are re-trieved from a warehouse. Pickers use automated guided vehicles on a fixed course collecting the items according to given customer orders. Those orders arrive through-out the study horizon, and this research focuses on the minimization of average turnover times.

[9] focus on the optimal number of customer orders that should be assigned to a batch such that the average turnover time is minimized. They employ a queuing network with two queues. In the first queue, customer orders arrive according to a Poisson process and batches are generated by means of the FCFS rule (First Come First Serve). If a particular number of customer orders are in the first queue, those orders are assigned to a batch and move onto the second queue. Those orders are released according to the availability of pickers.

A study of the average turnover time of a random customer order for a two-block layout is carried out by [10]. A corresponding model for all customer orders arriving during a particular time interval are assigned to batches in a two-block layout is presented by [11].

[12] explain an order picking area that is divided into several zones of identical size. The articles of each batch are picked up sequentially by zones. For this picking process, the researchers give an estimation of the average turnover times and observe that an optimal batch size exists.

[7] describes an on-line order batching problem in a walk-and-pick ware-house in which the completion times of all (dynamically arriving) customer orders (or the makespan) are to be minimized. The author also shows modifications of solution approaches for off-line order batching in order to deal with the on-line situation.

[5] introduces a GA for the order batching problem considering tra-versal and return routing policies. The proposed GA is tested on randomly generated instances and compared with the well-known savings algorithm. According to the author's extensive computational experiments, we can say that the proposed GA yields promising solutions in acceptable computation times.

The main characteristic in all this current research is the common representation of the solution. The authors employ discrete vectors where the number of elements equals the total number of orders to pick up and where each element contains an integer value that represents the batch to be assigned. We explained in the introduction section above that these algorithms try to preserve batches or sequences of batches that appear frequently in all members of the population in order to keep track and inherit these characteristics exhibited by the parents to the next generation. However, the traditional evolutionary operators used in current research may lose the characteristics mentioned.

4 EDA for the Order Batching Problem

Instead of using crossover and mutation operators of GA, the EDA produces offspring implicitly by using a probability model according to the knowledge of the obtained solution space during the search process. At the beginning of each generation, a subset of the population is determined and the obtained best solution is updated. Then,

the probability model is updated based on the sub-set mentioned and the offspring is generated by sampling the probability model. The EDA repeats the above process until a stopping condition is met.

4.1 Solution Representation

A solution to the order batching problem previously mentioned is expressed by the assignment of customer orders to batches, i.e., an order assignment vector represents a solution where the number of elements equals the total number of orders to pick up, where each element contains an integer value. This representation is shown in Figure 2 with 10 orders. Observe that the first batch consists of orders {5; 7; 9}; the second batch includes orders {1; 3; 4; 6} and the third batch incorporates order {2; 8; 10}.

		Customer Orders									
		1	2	3	4	5	6	7	8	9	10
Batch		2	3	2	2	1	2	1	3	1	3

Fig. 2. Solution vector.

4.2 Probability Model Proposed

The probability model is designed as a probability matrix, i.e., batch probability matrix P . The element p_{ji} of the probability matrix represents the probability that the batch j be used for the i customer order, where $j = \{1, \dots, m\}$ and $i = \{1, \dots, n\}$. The value of p_{ji} indicates the opportunity of a customer order on a certain batch. To build the probabilistic model or probability matrix P we need to compute the number of times that batch j appears for the customer i in the selected population. The result obtained previously is divided by the number of individuals considered in the selected population. Via sampling according to the probability matrix new promising individuals may be generated. For every position i , batch j is selected with the probability p_{ji} . If the batch j has already filled according to the device capacity, it means the assignment of the batch j has been finished. Then, the whole column $p_{j1}, p_{j2}, \dots, p_{jn}$ of probability matrix will be set as zero. This updating mechanism considers the previous assignments.

The probability model describes the characteristics exhibited by the parents as a distribution of the solution space. Therefore, the algorithm proposed does not need to hold members of the population to keep track and inherit the characteristics mentioned in each generation. The characteristics persist for all members of the population in the evolutionary progress by the probability model.

The major procedure of the EDA is listed as follows:

Step 1. Set the generation index $g = 0$. Initialize an initial population $S(0)$ of size M .

Step 2. Select a subset D from $S(g)$ of size N , where $N \leq M$.

Step 3. Establish the probabilistic model P which describes the distribution characteristics of D .

Step 4. Generate a set K of size M of new individuals by sampling P .

Step 5. A binary tournament is done to select the best individuals from $K \cup S(g)$ and assign them to the next generation $S(g+1)$.

Step 6. Let $g = g+1$. If $g < GN$, where GN is the maximum number of generations return to step 2. Otherwise, output the best solution in $S(g)$.

To describe the distribution of the solution space, the EDA tries to determine the relationship or interaction among variables of the problem as primary objective. The traditional evolutionary operators such as crossover and mutation are substitute by the probability model which is built with information on relationships and interactions among variables. The probability model on step 3 plays an important role in the EDA. Therefore, it should be well designed according to the characteristics of the considered problem. Step 4 makes another main difference between the EDA and other evolutionary algorithms. Sampling offspring from the probability matrix instead of using classical operators is the critical feature.

5 Results and Comparison

Three evolutionary algorithms have been proposed as benchmark for comparison with the EDA scheme. A GA based on [13] works with tournament selection. The “edge recombination operator” is used as a cross operator and a mutation operator changes batches among different positions. A TS based on [1] works with the operator “swap move”, meaning the interchanging of two customer orders from different batches. A tabu list is built on ten neighbors. A MA base on [14] works with the same operator for the local search phase.

To account for the stochastic nature of the order-picking warehouse, we ran 50 trials for all algorithms. We established a workload that contains different customer orders and differing types of articles required in a workday, replicating the order-picking process conditions to evaluate and find the minimum total completion time. Our experiments were based on a single-block warehouse with two cross aisles. It is assumed that there is one in the front and one in the back of the picking area. This layout was used by [1]. In this layout, the picking area consists of 900 articles and storage locations. The storage locations are arranged into 10 aisles. We further assume that an order picker walks through 10 storage locations in 30 seconds and they need 10 seconds to search and collect an article from a storage location. For the capacity of the picking device K , we assume three different values, namely 45, 75 and 100 articles. For the total number of orders n we consider 30, 60 and 90. The orders should arrive within a planning period of eight hours. The inter arrival times, i.e., the time between the arrival of order i and order $i+1$, are exponentially distributed with a parameter λ called arrival rate. Let $X(t)$ be the number of incoming orders in the time

interval $[0, t]$. In the case of exponentially distributed inter arrival times $E[X(t)] = \lambda \cdot t$ holds. In our numerical experiments we choose λ in a way that the expectation $E[X(t)]$ is equal to n for $t = 8[h]$. In summary, we use the following values for λ : for $n = 30$: $\lambda = 0.0625$, for $n = 60$: $\lambda = 0.125$, for $n = 90$: $\lambda = 0.1875$.

This dynamic case is characterized as an order picking environment in which the customer orders arrive stochastically over time, and only when an order has arrived the information becomes available of which articles and respective quantities the order is composed of. Under such conditions time window batching is prevalent. We set a fixed time window batching, i.e., all customer orders arriving during a particular time interval are assigned to batches by means of the probability model described above. The fixed time window batching for this tested scenario is set hourly. The unprocessed customer orders are input for building the initial population and the probabilistic model proposed in periods of one hour. The solution representation, i.e., the solution vector gets its dimension on the number of unprocessed customer orders confirmed previously. Therefore the batching decision is based on the known orders only. Once the batches are built by the sampling process, these are released one by one because we only have one picker to collect the parts in this tested scenario.

Table 1 details the minimum total completion time obtained for each case among all the trials.

Table 1. Comparison of results.

Capacity	Orders	Algorithms			
		GA	MA	TS	EDA
45	30	659	659	659	665
	60	840	784	834	784
	90	1926	1738	1818	1716
75	30	694	694	694	694
	60	1422	1329	1391	1328
	90	2294	2109	2231	2096
100	30	711	711	712	712
	60	1316	1214	1259	1214
	90	2011	1892	1981	1891

6 Discussion and Conclusions

On robustness, the algorithms utilized in this research are not able to handle invalid or unexpected inputs. These have not been encoded for specific users. This topic has not been considered in this research because it is not the main objective. However, the algorithm proposed can be modified in order to get a useful module for specific users in industry. On convergence and diversity, the algorithms used in this research keep diversity to incorporate specific operators such as the mutation operator in GA and TS in the evolutionary progress. Those operators are useful on permutation-based prob-

lems. On computational time and cost, these were not considered in this research because the algorithm proposed is currently in the prototype phase. Future research work would consider a module for users, and it should include computational time and cost aspects. On advantages and disadvantages of the proposed method, we can consider that it takes into account the relationship or interactions among variables of the problem as an advantage. For each generation, we know the probability that batch j was used for the i customer order. However, the probabilistic model used could be basic; it may be a disadvantage if we need to model higher interactions. On global optimum, note that for this problem, there are no known effective precise techniques and a comparison with an optimum solution is not possible. It is a characteristic of the on-line optimization topic. On computational complexity, the on-line order batching problem is as NP-hard. On feasibility and flexibility, on one hand all the algorithms used in this research were able to produce feasible solutions according to different constraints detailed in the problem statement section of this paper. It was not necessary to repair the solutions as other algorithms used for permutation-based problems. The proposed method considers the previous results in order to avoid unfeasible solutions. On the other hand, the algorithms utilized in this research are not flexible to handle new and unexpected customer orders. The proposed method is currently in the prototype phase for users. On efficiency and effectiveness, in this research the amount of resources used by all the algorithms was not considered, e.g., the requirement for high speed or for minimum memory usage were of no interest. On reliability and user friendliness, all the algorithms were tested in order to get reliability according to the on-line order batching characteristics. However, these are not industry-ready yet. On exploitative and exploration capability, all the algorithms used in this research keep exploitative and exploration capability to incorporate specific operators such as cross and mutation operator in GA and TS in the evolutionary progress.

Based on the experimental results shown, we deduce that the EDA can be an efficient algorithm to handle different order-picking conditions such as capacity and customer orders. When the amount of orders increase the EDA has an acceptable performance according to the results detailed above. The probability model is useful in order to preserve batches or sequences of batches that appear frequently in all members of the population. The EDA is able to describe the characteristics exhibited by the parents in each generation. Finally, this research contributes by using an EDA as an optimization method for any order-picking process. Future research work will use higher probabilistic models in order to model higher interactions or relationships between variables of the order-picking performance.

Acknowledgements. We would like to express our gratitude to Elizabeth O'Shaughnessy for reviewing the manuscript.

References

1. Henn, S.: Metaheuristics for the order batching problem in manual order picking systems. *Ottovon-Guericke-University Magdeburg, Faculty of Economics and Management, Verband der Hochschullehrer für Betriebswirtschaft e.V.*, 3(1), 82-105 (2009)

2. de Koster, M., Le-Duc, T., Roodbergen, K.: Design and Control of Warehouse Order Picking: A Literature Review. *European Journal of Operational Research*, 182(2), 481–501 (2007)
3. Wäscher, G.: Order Picking: A survey of planning problems and methods. In Dyckhoff, H., Lackes, R., Reese J. (eds.) *Supply Chain Management and Reverse Logistics*, pp. 323–347. Springer, Berlin (2004)
4. Bozer, Y., Kile, J.: Order Batching in Walk-and-Pick Order Picking Systems. *International Journal of Production Research*, 46(7), 1887–1909 (2008)
5. Öncan, T.: A Genetic Algorithm for the order batching problem in low-level picker-to-part warehouse systems. In: *Proceedings of the International MultiConference of Engineers and Computer Scientists IMECS 2013*, vol. I. Hong Kong (2013)
6. Mühlenbein, H., Paaß, G.: From recombination of genes to the estimation of distributions: I. binary parameters. In: Voigt, H., Ebeling, W., Rechenberg, I., Schwefel H. (eds.) *Parallel Problem Solving from Nature PPSN IV*, pp. 178–187, Springer, Berlin (1996)
7. Henn, S.: Algorithms for on-line order batching in an order-picking warehouse. In: *Proceedings of the 3rd International Conference on Information Systems, Logistics and Supply Chain ILS 2010*, Casablanca: Business Process Consulting (2010).
8. Kamin, N.: On-line optimization of order picking in an automated warehouse. Ph.D. Dissertation, Shaker Verlag, Aachen (1998)
9. Chew, E., Tang, L.: Travel time analysis for general item location assignment in a rectangular warehouse. *European Journal of Operations Research*, 112(3), 582–597 (1999).
10. Le-Duc, T., de Koster, R.: Travel time estimation and order batching in a 2-block warehouse. *European Journal of Operations Research*, 176(1), 374–388 (2007)
11. van Nieuwenhuysse, I., de Koster, R.: Evaluating order throughput time in 2-block warehouses with time window batching. *International Journal of Production Economics*, 121, 654–664 (2009)
12. Yu, M., de Koster, R.: The impact of order batching and picking area zoning on order picking system performance. *European Journal of Operational Research*, 198(2), 480–490 (2009)
13. Whitley, D., Starweather, T., Shaner, D.: The traveling salesman and sequence scheduling: quality solutions using genetic edge recombination. In: Davis L. (ed.) *Handbook of genetic algorithms*, pp. 350–372, New York (1990)
14. Moscato, P.: On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts: Towards Memetic Algorithms. Caltech Concurrent Computation Program (report 826) (1989)

Mecanismo de transformación de diagramas UML de casos de uso a código WSCDL

Isaac Machorro Cano¹, Daniel Eduardo Díaz Navarro², Giner Alor Hernández³,
Mónica Guadalupe Segura Ozuna¹, Lisbeth Rodríguez Mazahua³

¹ Universidad del Papaloapan (UNPA), Tuxtepec, Oaxaca,
México

²Instituto de Estudios Superiores de Oaxaca A.C. (IESO), Oaxaca, Oaxaca,
México

³ Instituto Tecnológico de Orizaba (ITO), Orizaba, Veracruz,
México

{imachorro, msegura}@unpa.edu.mx {dandiaznavarro}@gmail.com
{galor, lrodriguez}@itorizaba.edu.mx

Resumen. Los diagramas UML de Casos de Uso definen el comportamiento de un sistema a través de las interacciones de los actores y los procesos, además determinan los requisitos funcionales del sistema. Adicionalmente, las empresas están diseñando sus procesos de negocios mediante la tecnología de servicios Web y UML se ha posicionado como un lenguaje estándar de modelado para el desarrollo de aplicaciones. Por otra parte, con la utilización de las tecnologías de servicios Web y UML, es posible modelar nuevos procesos de negocios más complejos, los cuales requieren de mecanismos para lograr una mejor interoperabilidad a nivel empresarial. Estos mecanismos se logran a través de lenguajes composicionales en servicios Web como WS-CDL el cual se basa en XML. Este trabajo presenta un mecanismo de transformación que permite la generación de código WSCDL a partir de diagramas UML de Casos de Uso y un caso de estudio para ejemplificar su funcionalidad.

Palabras clave: mecanismo de transformación, WS-CDL, diagramas de casos de uso.

1. Introducción

Los diagramas UML (*Unified Modeling Language*) de Casos de Uso (DCU) permiten describir las funciones de un sistema desde el punto de vista de sus interacciones con él, sin detallar la descripción de la implementación y determinan los requisitos funcionales del sistema [1]. Los DCU son también la base para los diagramas relacionados de componente y despliegue. Dentro de los elementos que conforman los DCU se tienen: a) Actores.- Se le llama así al usuario, cuando desempeña ese papel con respecto al sistema. Estos llevan a cabo los casos de uso, un mismo actor realiza mu-

chos casos de uso; b) Caso de Uso.- Describe una tarea que realiza el sistema, representa también cómo el sistema colabora con el actor para realizar al menos una tarea; c) Relaciones entre Casos de Uso.- Entre los casos de uso se establecen tres tipos de relaciones: (1) Extends: es similar a un caso de uso, al aplicar esta relación implica que un caso de uso difiere en su comportamiento dependiendo de ciertas circunstancias. Incorpora implícitamente el comportamiento de otro caso en el lugar especificado indirectamente por este caso; (2) Include: incorpora explícitamente el comportamiento de otro caso en algún lugar de su secuencia; (3) Generalización: hereda el comportamiento y significado de otro caso de uso [2]. Así mismo el creciente desarrollo y utilización de Internet en el ámbito comercial, ha propiciado que este sea el lugar donde la mayoría de las organizaciones ponen a disposición sus modelos comerciales, ayudándose de los servicios Web. El W3C (*Consortium World Wide Web*), define a los servicios Web (SW) como un sistema diseñado para soportar la interacción máquina a máquina sobre una red que cuenta con una interfaz descrita en un formato que se procesa en una máquina WSDL (*Web Services Description Language*). La idea principal es que los servicios tradicionales se manejen como un SW, además los SW son sustancialmente un nivel de infraestructura existente entre modelos de componentes [3]. Para el desarrollo de estos SW existen muchas herramientas y lenguajes que facilitan la creación de dichos servicios, tales como XML (*eXtensible Markup Language*), SOAP (*Simple Object Access Protocol*), UDDI (*Universal Description, Discovery and Integration*) y WSDL [4].

Por otra parte, con la utilización de las tecnologías de SW y UML, es posible modelar nuevos procesos de negocios más complejos, los cuales requieren de mecanismos para lograr una mejor interoperabilidad a nivel empresarial. Estos mecanismos se logran a través de lenguajes composicionales en SW tales como WSCI (*Web Service Choreography Interface*), BPML (*Business Process Modeling Language*), WSFL (*Web Service Flow Language*), XLANG (*Web Services for Business Process Design*) y los más utilizados como BPEL (*Business Process Execution Language for Web Services*) y WS-CDL (*Web Services Choreography Description Language*). WSCDL es mejor y más poderoso que BPEL, debido a que se centra básicamente en la descripción de la coreografía de los SW, además proporciona una definición de los formatos de información que se intercambia por todos los participantes, mientras que BPEL solo lo realiza para un participante. Así mismo, WSCDL proporciona el intercambio global de mensajes entre los participantes sin un punto de vista específico, mientras que BPEL lo realiza desde el punto de vista de solo un participante. WSCDL ofrece reglas que utiliza cada participante para estimar el estado de la coreografía y deducir cuál podría ser el siguiente intercambio de mensajes y BPEL especifica reglas que se ejecutan para deducir las actividades posteriores a realizar, una vez que se calcula la regla, el tiempo de ejecución de la orquestación ejecuta las actividades correspondientes. Además WSCDL ofrece un modelo para la especificación del mejor entendimiento del intercambio de mensajes y BPEL no tiene tal concepto. Tomando en cuenta lo descrito anteriormente, en este trabajo se presenta un mecanismo de transformación que permite la generación de código WSCDL a partir de diagramas UML de Casos de Uso y un caso de estudio para ejemplificar la funcionalidad del mecanismo de transformación propuesto; en donde se realiza un mapeo de DCU a las etique-

tas de WSCDL, se genera su correspondiente documento XMI (*XML Metadata Interchange*) y se desarrolla un conjunto de clases en Java para la implementación y generación de código WSCDL.

Este trabajo se estructura como sigue, en la sección 2 se presenta la estructura y descripción de un documento WSCDL, en la sección 3 se presenta el trabajo relacionado, en la sección 4 se presenta el mecanismo de transformación de DCU a código WSCDL, en la sección 5 se presenta el caso de estudio de la búsqueda del proveedor que ofrezca el precio más bajo de un libro y finalmente en la sección 6 se presentan las conclusiones, el trabajo a futuro y los agradecimientos.

2. WSCDL

WS-CDL describe colaboraciones interoperables entre sus participantes. Estas colaboraciones tienen un entorno común, donde hay una serie de reglas definidas y restricciones que se deben seguir para el intercambio de la información. Diferentes características se incluye en el lenguaje para proveer una reflexión completa y exacta de qué es necesario para modelar una coreografía en particular y dependiendo de lo que requieran, algunas características del lenguaje se determina si son completamente necesarias o no [5].

Tabla 1. Estructura de un documento WSCDL.

Etiqueta WSCDL	Descripción
<package>	
name="NCName"	Nombre del documento WS-CDL
author="xsd:string"	Autor del documento WS-CDL
version="xsd:string"	
targetNamespace="uri"	
xmlns="http://www.w3.org/2005/10/cdl">	
<informationType/>	Detalla el tipo de información de la variable o el token de referencia.
<token/>	Es un alias que se usa para hacer referencia a una variable o partes de ella.
<tokenLocator/>	Localiza un token dentro de un mensaje
<roleType/>	Refiere el comportamiento potencial que un participantType puede mostrar en una interacción
<relationshipType/>	representa las relaciones entre los roles
<participantType/>	Agrupar aquellas partes del comportamiento que debe de implementar por la misma entidad lógica.
<channelType/>	Representa un canal donde se realiza una colaboración entre "participantTypes" puntualizando donde sucede el intercambio de información.
Choreography-Notation	Este elemento especifica una colaboración entre "participantTypes" que intercambian información dentro de la coreografía
<package	

La estructura de un documento WS-CDL se muestra en la Tabla 1, en donde se observan los elementos a utilizar en una coreografía así como una breve descripción de cada uno de ellos. WS-CDL es una descripción y no un lenguaje ejecutable, se utiliza para describir colaboraciones de servicios notables; esto se entiende como un protocolo

lo comercial. Además WSCDL es un lenguaje basado en XML que se usa para describir la conducta colaborativa y común entre múltiples servicios que necesitan interactuar entre si en un orden para lograr alguna meta.

En la siguiente sección se presentan algunos trabajos relacionados con esta propuesta, en donde se hace uso de diagramas UML, WSCDL u otros lenguajes composicionales, SW y del lenguaje de programación Java entre otros aspectos relacionados con este trabajo.

3. Trabajos relacionados

En [6], se propone un nuevo enfoque basado en la planeación, incluyendo la compilación de contingencias, estado de acciones, análisis de dependencia y comunicación que convierten automáticamente una tarea de composición a una especificación distribuida de la coreografía. Por otra lado en [7] se propone un simple lenguaje llamado *Chor c*, el cual es un sub-idioma de interacción para modelar el aspecto del canal de la coreografía del servicio, además se estudian algoritmos para la comprobación estática de la coreografía del servicio y la generación del canal. Por otra parte en [8], se presentan cinco teoremas los cuales son útiles para trabajar con el modelado de diagramas de secuencia y establecer una teoría del modelo orientado a aspectos o un enfoque de la transformación de los modelos. En [9] se extraen formalmente especificaciones basadas en WSCDL en un proceso de comunicación secuencial soportando un enfoque formal para comprobar los modelos de SW, esto a través de un ejemplo PAT adaptado mediante la realización de un mapeo de WSCDL a CSP con el objetivo de obtener las descripciones de CSP de la coreografía de SW. Así mismo en [10], se presenta una revisión de trabajos sobre la integración de las representaciones de procesos de negocios y la tecnología de SW, identificando la necesidad de desarrollar propuestas de transformación entre modelos, representar mapeos de una manera formalizada y una estructura de ejecución común.

Por otra parte en [11], se presenta una metodología para el desarrollo de los SW basados en diagramas de secuencia en UML 2.0 como especificaciones de partida en la composición de SW, estos diagramas se traducen en lenguaje WS-CDL. Además en [12], se propone un método basado en CPN que detecta los errores en la fase de diseño y asegura la exactitud de la composición de servicios para un análisis estructural estático y un análisis de comportamiento dinámico, presentando un ejemplo del mapeo entre los elementos de la composición de SW y CPN. Así mismo en [13], se presenta la herramienta WSCVT, la cual utiliza Java para verificar un algoritmo RBSR, en donde el enfoque propuesto se verifica utilizando la herramienta WSCVT y los resultados revelaron un mejor rendimiento en términos de evitar problemas de escasez y accesibilidad. Además se presenta una traducción de los términos de WSCDL en relación a los términos de un RBA. Por otra parte en [14], se presenta una verificación de los SW compuestos utilizando la interface de autómatas, en donde inicialmente la composición se realiza mediante el uso de BPEL, el cual se convierte en autómatas de interfaz dentro de Promela y posteriormente verificadas utilizando la herramienta llamada SPIN. Además en [15], se presenta una construcción de SW utilizando Re y

autómatas restringidos; igualmente proponen una estructura que tomando como entrada la descripción del comportamiento del servicio, sus interfaces WSDL y su interacción en Reo, genera el correspondiente código en Java para orquestar los servicios.

En la siguiente sección se presenta el mecanismo de transformación de DCU a código WSCDL en donde se describe un mapeo de un DCU a sus correspondientes etiquetas de WSCDL, la descripción de las etiquetas de la coreografía del código en WSCDL y la descripción de las clases en Java desarrolladas.

4. Mecanismo de transformación

Para el mecanismo de transformación propuesto en este trabajo que consiste en la transformación de DCU a su correspondiente código en WS-CDL, previamente se han realizado otras pruebas con los diagramas UML de Actividades y de Secuencia, realizando un mapeo de cada diagrama a las etiquetas en WSCDL y generando a partir de los diagramas su correspondiente código en WSCDL, mediante diversos escenarios presentados [16]. Por lo tanto, en esta propuesta se utilizó un DCU que representa el caso de estudio de la “Búsqueda del proveedor que ofrezca el precio más bajo de un libro”. La Fig. 1 muestra el mapeo de las etiquetas correspondientes al lenguaje WS-CDL y su correspondiente descripción relacionada al DCU en donde:

1. La etiqueta `<relationshipType>` se genera principalmente al pasar de un rol a otro. Por ejemplo al pasar del rol WS-CDL al rol BPEL4WS.
2. La etiqueta `<informationType>` y `<variables>` se generan con la información que se va obteniendo de cada caso de uso, como por ejemplo “Ingresar dato del libro”, “Buscar por autor”, entre otros.
3. La etiqueta `<exchange>` se genera al haber intercambio de información entre el caso de uso “Buscar Libro”, que está dentro del rol WS-CDL al caso de uso “Buscar precio y proveedor del libro” del rol BPEL4WS.
4. La etiqueta `<token>` y `<tokenLocator>` se crean cuando se localizan y definen los alias a los casos de uso, esto es la información de cada uno, por ejemplo el caso de uso “Buscar libro”.
5. La etiqueta `<roleType>` se genera cuando existe por lo menos un rol dentro del DCU, en este caso se cuentan con tres roles: WS-CDL, BPEL4WS y WSDL.
6. La etiqueta `<interation>` se obtiene cuando se pasa de un caso de uso a otro.

Finalmente en la etiqueta `<choreography>` se agrupan todas las etiquetas anteriormente mencionadas e incluimos la etiqueta `<sequence>` que representa la secuencia de intercambio de información entre los casos de uso, esto para realizar su correspondiente coreografía en WSCDL.

Una vez identificadas las etiquetas de WSCDL en el diagrama, el código generado en WSCDL se muestra en la Fig. 2, en donde se observa que:

1. El inicio de la coreografía es mediante la etiqueta `<choreography>` en donde se indica el nombre y descripción del caso de estudio.

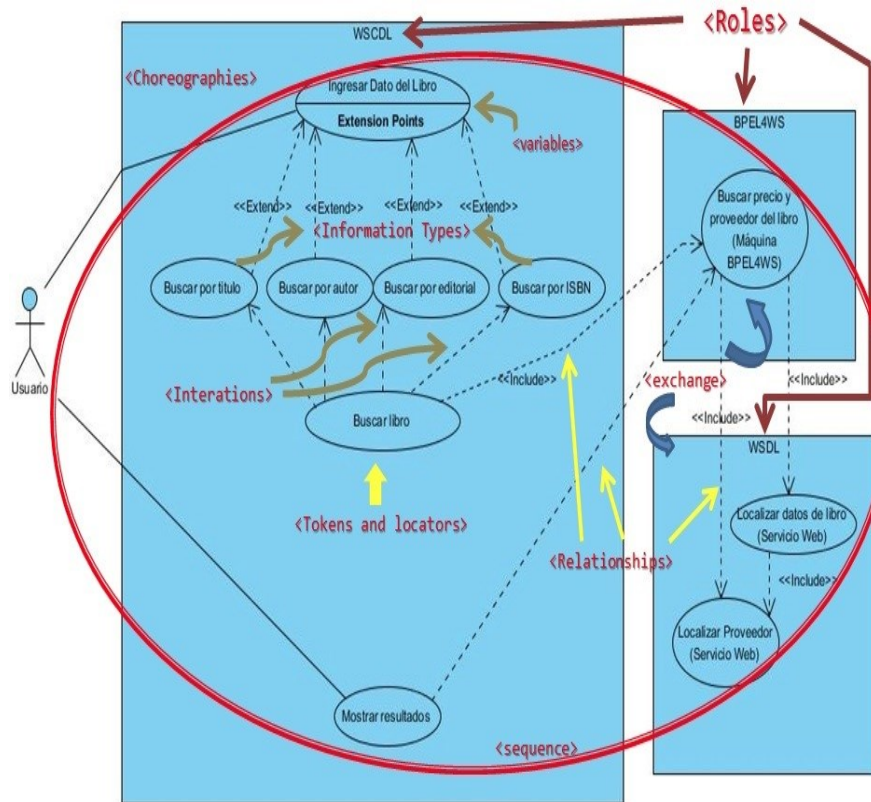


Fig. 1 Diagrama UML de Casos de Uso

2. Posteriormente en la etiqueta `<variableDefinitions>` se indican todas las variables a utilizar obtenidas de los diferentes casos de uso, indicando en cada una de ellas para que se utilizará.
3. Además a través de la etiqueta `<sequence>`, se indica que la coreografía es secuencial. Después indicamos con la etiqueta `<interaction>` el tipo de caso de uso, dentro del cual se establece la operación y su correspondiente canal.
4. Por otra parte en las etiquetas `<participate>` `<relationshipType>` se indica el tipo de relación entre los participantes de la coreografía a través de los roles, para este caso los roles son “WSCDL”, “BPEL4WS” y “WSDL”.
5. Para la comunicación entre los diferentes casos de usos se utiliza la etiqueta `<exchange>`, en donde para realizar el envío de mensajes se realiza mediante la etiqueta `<send>` y la recepción de los mismos con la etiqueta `<receive>`, así mismo el tipo de acción que se desea realizar a través de las palabras reservadas “response” o “request” según sea el caso.

Finalmente para el cierre de cada etiqueta abierta se antepone el símbolo “/” según sea el caso; la coreografía del caso de estudio propuesto finaliza con la etiqueta `</choreography>`.

```

<choreography
name="B\u00fasqueda del proveedor que ofrezca el precio m\u00e1s bajo de un
libroChoreography" root="true">
<descriptiontype="documentation">TheChoreographyfor B\u00fasqueda del proveedor
que ofrezca el precio m\u00e1s bajo de un libro</description>
<relationship type="tns:BPPEL4WS2WSCDL"/>
<variableDefinitions>
<variable channelType="tns:BPPEL4WS2WSCDLChannel"
name="BPPEL4WS2WSCDL" roleTypes="tns:BPPEL4WSRole tns:WSCDLRole">
<description type="documentation">Channel Variable</description>
</variable>
<variable informationType="tns:BuscarlibroType"
name="Buscarlibro" roleTypes="tns:BPPEL4WSRole tns:WSCDLRole">
<description type="documentation">Request Message</description>
</variable>
<variable
informationType="tns:Buscar precio y proveedor del libro (M\u00e1s bajo de un
BPPEL4WS)Type"
name="Buscar precio y proveedor del libro (M\u00e1s bajo de un BPPEL4WS)"
roleTypes="tns:BPPEL4WSRole tns:WSCDLRole">
<description type="documentation">Response Message</description>
</variable>
</variableDefinitions>
<sequence>
<interaction channelVariable="tns:BPPEL4WS2WSCDL"
name="Interaction1" operation="Operation1">
<description type="documentation">Interaction1</description>
<participate fromRoleTypeRef="tns:BPPEL4WSRole"
relationshipType="tns:BPPEL4WS2WSCDL" toRoleTypeRef="tns:WSCDLRole"/>
<exchange action="request"
informationType="tns:BuscarlibroType" name="Buscarlibro">
<description type="documentation">Request Message Exchange</description>
<send variable="cdl:getVariable('Buscarlibro','','')"/>
<receive variable="cdl:getVariable('Buscarlibro','','')"/>
</exchange>
<exchange action="respond"
informationType="tns:Buscar precio y proveedor del libro (M\u00e1s bajo de un
BPPEL4WS)Type" name="Buscar precio y proveedor del libro (M\u00e1s bajo de un
BPPEL4WS)">
<description type="documentation">Response Message Exchange</description>
<send variable="cdl:getVariable('Buscar precio y proveedor del libro
(M\u00e1s bajo de un BPPEL4WS)',','')"/>
<receive variable="cdl:getVariable('Buscar precio y proveedor del libro
(M\u00e1s bajo de un BPPEL4WS)',','')"/>
</exchange>
</interaction>
</sequence>
</choreography>

```

Fig. 1. Código WSCDL

La implementación del mecanismo de transformación se llevó a cabo siguiendo los tres pasos que a continuación se describen:

Paso 1.- Se modeló el DCU del caso de estudio de la búsqueda del proveedor que ofrezca el precio más bajo de un libro en la herramienta de modelado UML *Visual Paradigm for UML 8.0 Enterprise Edition* la cual es una herramienta que colabora en el diseño de software.

Paso 2.- Una vez generado el diagrama con la herramienta de modelado, se exportó el DCU a un documento XMI, el cual es un estándar de la OMG (*Object Management Group*) para el intercambio de información de metadatos utilizando XML. En la Tabla 2, se indica la relación de los elementos de un documento XMI con respecto a las etiquetas del código en WSCDL.

Tabla 2. Relación entre XMI y WSCDL

Documento XMI	Código WSCDL
<uml:Diagram>	<package>
<ownedMember>	roleType
<uml:Model>	<participantType>
	<relashionsipType>
<include>	<exchange>
addition (origen)	<send>
	response request
xmi:id (destino)	<recibe>
	response request
<extends>	<exchange>
extendedCase (origen)	<send>
	response request
xmi:id (destino)	<recibe>
	response request
<uml:Dependency>	<exchange>
client (origen)	<send>
	response request
supplier (destino)	<recibe>
	response request
<ownedMember>	<informationType>
	<token>
	<tokenLocator>

<choreography>
 <variableDefinitions>
 <sequence>
 <interaction>
 <participate>
 <relationshipType>
 <exchange>
 <send>
 response
 request
 <recibe>
 response
 request
 <exchange>
 .
 .
 .
 <exchange>
 <send>
 response
 request
 <recibe>
 response
 request
 <exchange>
 <choreography>

Paso 3.- Para la generación del código se desarrollaron cuatro clases en el lenguaje de programación Java, el cual cuenta con JAXP (*Java API for XML Processing*) una API (*Application Programming Interface*) que sirve para procesar archivos XML. JAXP aprovecha los estándares de las API s SAX (*Simple API for XML*) y DOM (*Document Object Model*), de tal manera que se analizan los datos como secuencia de eventos, de igual forma, al ser diseñado para ser flexible, JAXP permite usar cualquier analizador que trabaje con XML. SAX igualmente permite interpretar un archivo que utiliza XML detectando cuando empieza y termina un elemento del documento, además de comprobar que el documento este bien formateado. SAX no carga todo el documento en memoria solo lo que se está analizando, por lo cual es muy eficiente en cuanto al tiempo y memoria que se usan en el análisis. Así mismo para generar el código WSCDL, se utilizó la API DOM la cual describe la estructura para un documento XML, con ella podemos navegar en el documento e insertar nodos. La especificación que hace el W3C define a DOM como una interfaz independiente de cualquier plataforma que permite a programas acceder y actualizar dinámicamente la estructura del documento. En la Figura 3 se muestran las cuatro clases que se desarrollaron para la realización de este proyecto, en donde:

1. La **clase FramePrincipal**.- Dibuja un Frame para seleccionar el documento XMI, invoca a la clase “CasoDeUso” y visualiza el documento WSCDL.
2. La **clase CasoDeUso**.- Invoca a las clases “ObtenerEtiquetas” y “GenerarWSCDL”.
3. En la **clase ObtenerEtiquetas**.- Se definen los métodos para obtener los elementos del código WS-CDL, a través del archivo XMI. Esta clase cuenta los métodos: `getNameDiagram()` obtiene el nombre del diagrama del archivo XMI, `getAuthorDiagram()` obtiene el autor del diagrama de casos de uso, `getRole()` extrae los diferentes grupos que interactúan dentro del diagrama, `getCasosDeUso()` obtiene la información de los casos de uso que se encuentran dentro de cada grupo, `getRelations()` obtiene las relaciones que existen entre los casos de uso y el método `getActionCasos()` extrae que acción representa cada caso de uso, ya sea un “request” o un “response” hacia otro caso de uso.
4. La **clase GenerarWSCDL**.- Cuenta con el método `createWSCDL()` el cual genera el documento WS-CDL con la información obtenida en la clase “ObtenerEtiquetas” del archivo XMI.

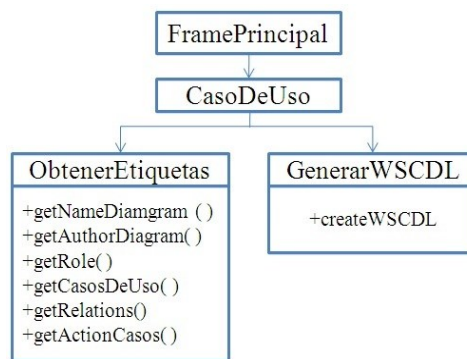


Fig. 2. Clases en Java

En este mecanismo de transformación aplicado al caso de estudio propuesto, se observa la generación de código WSCDL a partir de un diagrama UML de Casos de Uso, mostrando el mapeo del diagrama UML de Casos de Uso a los elementos del código en WSCDL, la relación entre el documento XMI con el código WSCDL, así como su correspondiente implementación.

En la siguiente sección se presenta un caso de estudio de la búsqueda del proveedor que ofrezca el precio más bajo de un libro en el cual se basa este mecanismo de transformación y que sirvió para la validación del mismo.

5 Caso de estudio: “Búsqueda del proveedor que ofrezca el precio más bajo de un libro”

Para la validación del mecanismo de transformación de DCU a código WSCDL se planteó el siguiente escenario: Un usuario desea saber el proveedor que ofrece el pre-

cio más bajo de un determinado libro. Para esto, recurre a un servicio que le oriente en su búsqueda y le proporcione diversos resultados. De los proveedores propuestos, el usuario decidirá cuál de ellos es el que ofrece el precio más bajo del libro.

Una vez planteado el caso de estudio, se realizó el modelado del correspondiente DCU en donde como se muestra en la Fig. 1 se identifican los tipos de relaciones, los diferentes Casos de Uso y a qué rol pertenecen. Además se inicia con el caso de uso “Ingresar Dato del Libro” en donde se realiza la búsqueda del mismo a través de los casos de uso “Buscar por título”, “Buscar por autor”, “Buscar por editorial” o “Buscar por ISBN” según sea el caso para pasar al caso de uso “Buscar libro” ubicados en el sistema WSCDL. El caso de uso “Buscar libro” se comunica con el caso de uso “Buscar precio y proveedor del libro (Máquina BPEL4WS)” ubicado en el sistema BPEL4WS en donde este realiza la petición de los datos correspondientes a los casos de uso “Localizar Proveedor (Servicios Web)” y “Localizar datos del libro (Servicio Web)” pertenecientes al sistema WSDL. Posteriormente se envía dicha información al caso de uso “Buscar precio y proveedor del libro (Máquina BPEL4WS)” para finalmente presentar los resultados de la búsqueda al usuario a través del caso de uso “Mostrar resultados” para que él tome la decisión de la compra del libro de acuerdo al proveedor que ofrezca el precio más bajo.

Después de modelado y descrito el funcionamiento del caso de estudio en el DCU, generamos a partir del diagrama el correspondiente archivo XMI, del cual obtenemos la representación de los elementos del lenguaje WS-CDL como se describe a continuación:

- Del documento XMI obtenemos el contenido de la etiqueta `<uml:Diagram>` que representa la etiqueta `<package>` en el documento WS-CDL.
- Dentro de la etiqueta `<ownedMember>` en el documento XMI, se encuentra la etiqueta `<uml:Model>` la cual representa la etiqueta “roleType” en WSCDL de donde se obtiene el elemento “participantType”, además de la etiqueta “relationshipType” ya que esta se da cuando dos roleType se comunican entre sí.
- Para obtener el elemento informationType en WSCDL, se busca la etiqueta `<ownedMember>` en el documento XMI, el cual debe de estar dentro de otra etiqueta con el mismo nombre, a partir de este elemento podemos extraer los elementos token y tokenLocator de WSCDL.
- El elemento choreography en WSCDL, se obtiene a partir de los elementos anteriormente mencionados así mismo se genera el elemento `<interaction>` en donde si existe un intercambio de información entre dos roleType se crea también la etiqueta `<exchange>`. En el documento XMI este intercambio lo obtenemos de diferentes etiquetas, una es `<include>` con su elemento “addition” que representa el origen y “xmi:id” que es el destino, de igual forma con la etiqueta `<extends>` y su atributo “extendedCase” que es el origen y el destino es “xmi:id”, y la última forma de obtener la etiqueta `<exchange>` es con la etiqueta `<uml:Dependency>` dentro del XMI, siendo el origen el atributo “client” y el destino el atributo “supplier”.

Una vez identificados y relacionadas las etiquetas del documento XMI con las etiquetas de WSCDL, la clase “ObtenerEtiquetas” lee el archivo XMI y guarda la información descrita anteriormente a través sus métodos para posteriormente pasar a la

clase "GenerarWSCDL" donde con dicha información se procede a generar el código en WS-CDL. Finalmente, se obtiene el código WS-CDL el cual es visualizado en cualquier editor o navegador que soporte XML.

En la siguiente sección se presentan las conclusiones a la que se llegaron al desarrollar el mecanismo de transformación, fundamentalmente al realizar el mapeo y al generar las clases en Java para la generación de código WSCDL, se plantea el trabajo a futuro y se presentan los agradecimientos.

6. Conclusiones y trabajo a futuro

Los SW son una plataforma para mostrar la interoperabilidad de servicios de las aplicaciones utilizando como estándar UML, el cual también se utiliza para la descripción de procesos de negocios. En este contexto los diagramas UML de Casos de Uso describen el comportamiento de un sistema visto desde el usuario, además estos diagramas determinan los requisitos funcionales del sistema y son importantes para la visualización, especificación, documentación del modelo estructural y construcción de un sistema. Además describen de una forma gráfica la composición de un SW, por lo cual es más fácil el entendimiento de su funcionalidad. Por otra parte podemos describir al lenguaje WS-CDL como punto de partida para la generación de la coreografía de los SW, mostrando la secuencia de mensajes entre las distintas partes y fuentes, gracias a que este lenguaje se basa en XML. En este trabajo se presentó un mecanismo de transformación que permite la generación de código WSCDL a partir de diagramas UML de Casos de Uso basado en el caso de estudio de la búsqueda del proveedor que ofrezca el precio más bajo de un libro lo cual permitió validar el mecanismo de transformación presentado.

Finalmente el trabajo a futuro consiste en analizar otros casos de estudio para validar y fortalecer el mecanismo de transformación de DCU a código WSCDL propuesto. Además se considera desarrollar adicionalmente bibliotecas de funciones en Java para generar código WSCDL a partir de diagramas UML de componentes y despliegue como continuación de este trabajo.

Agradecimientos. Los autores agradecen al Programa de Fortalecimiento de la Calidad en Instituciones Educativas (PROFOCIE 2014), al Programa para el Desarrollo Profesional Docente (PRODEP), al Consejo Nacional de Ciencia y Tecnología (CONACYT) y al Tecnológico Nacional de México por el apoyo otorgado para la realización de esta investigación.

Referencias

1. Cook, S.: Looking back at UML. *Software & Systems Modeling*, Vol. 11, Issue 4, pp. 471–80, Springer-Verlag (2012)
2. Falgueras, C. B.: *Ingeniería de software*. pp. 71–107, Editorial UOC (2003)
3. Bittner, K., Spence, L.: *Use Case Modeling*. pp. 3–18, Pearson Education (2003)

4. Arboleda, C. L. M.: Servicios WEB: Distribución e integración. Vol. 2, S & T Sistemas y Telemática (2004)
5. Ross, T. S., Fletcher, T.: Web Services Choreography Description Language: Primer W3C Homepage, <http://www.w3.org/TR/2006/WD-ws-cdl-10-primer-20060619/> (2014)
6. Zou, G., Gan, Y., Chen, Y., Zhang, B., Huang, R., Xu, Y., Xiang, Y.: Towards automated choreography of Web services using planning in large scale service repositories. *Applied Intelligence*, Vol. 41, Issue 2, pp. 383–404, Springer US (2014)
7. Yang, H., Cai, C., Peng, L., Zhao, X., Qiu, Z., Qin, S.: Algorithms for checking channel passing in web services choreography. *Frontiers of Computer Science*, Vol. 7, Issue 5, pp. 710–728, Springer Berlin Heidelberg (2013)
8. Gronmo, R., Kobro, R. R., Moller, P. B.: Confluence of aspects for sequence diagrams. *Software & System Modeling*, Volume 12, Issue 4, pp. 789–824, Springer Berlin, Heidelberg (2011)
9. Dong, X., Zhou, L., Wei-min, L., Bo-feng, Z.: Model checking web services choreography in process analysis toolkit. *Journal of Shanghai University*, Vol. 14, Issue 1, pp. 45–49, Shanghai University and Springer-Verlag (2010)
10. Grolinger, K., Capretz, M. A. M., Cunha, A., Tazi, S.: Integration of business process modeling and Web services: a survey. *Service Oriented Computing and Applications*, Vol. 8, Issue 2, pp. 105–128, Springer (2013)
11. Cambroneró, M. E., Ruiz, V. V., Martínez, E.: Design and Generation of Web Services Choreographies with Time Constraints. *J. UCS*, Vol. 17, Issue 13, pp. 1800–1829 (2011)
12. Tian, B., Gu, Y., Formal Modeling and Verification for Web Service Composition. *Journal of software*, Vol. 8, No. 11, Academy publisher (2013)
13. Danapaquiame, N., Ilavarasan, E.: Corroboration Strategy for Web Services Choreography using Revise Buchi Automata. *International Journal of u- and e- Service, Science and Technology*, Vol. 6, No. 5, pp 115–132 (2013)
14. Mei, J., Miao, H., Chen, Y., Gao, H.: Verifying Web Service composition based on Interface Automata using SPIN. *Journal of Digital Content Technology and it's applications*, Vol. 4, No. 8 (2011)
15. Jongmans, S. T. Q., Santini, F., Sargolzaei, M., Arbab, F., Afsarmanesh, H.: Orchestrating web services using Reo: from circuits and behaviors to automatically generated code. *Service Oriented Computing and Applications*, Vol. 8, Issue 4, pp. 277–297, Springer (2013)
16. Alor Hernandez, G., Machorro Cano, I., Gomez, J. M., Cruz Ahuactzi, J., Posada Gomez, R., Mencke, M., Juarez Martinez, U.: Mapping UML Diagrams for generating WS-CDL code. In: *Third International Conference on Digital Society (ICDS)*, IEEE (2009)

Factores predisponentes en relajación residual neuromuscular

María Dolores Torres Soto¹, Aurora Torres Soto², María de la Luz Torres Soto³,
Leticia Bermudez Rosales³, Eunice E. Ponce de León Sentí²

¹ Universidad Autónoma de Aguascalientes, Departamento de Sistemas de Información,
Aguascalientes, México

² Universidad Autónoma de Aguascalientes, Departamento de Ciencias de la Computación,
Aguascalientes, México

³ Centenario Hospital Miguel Hidalgo,
Aguascalientes, México

mdtorres@correo.uaa.mx, {atorres, eponce}@correo.uaa.mx,
{tishats, letybermudez3}@hotmail.com

Resumen. Este artículo muestra los resultados del seguimiento y análisis de 210 pacientes que ingresaron a la Unidad de Cuidados Post-anestésicos del Centenario Hospital Miguel Hidalgo de Abril a Septiembre de 2014 en la ciudad de Aguascalientes. La herramienta empleada para la identificación de los factores predisponentes de relajación residual neuromuscular fue un algoritmo genético hibridizado con los conceptos de testor y testor típico, así como la integración de operadores genéticos que permiten la generación y mejora de individuos prometedores durante el proceso. La relajación residual neuromuscular es la condición clínica determinada por la persistencia de los efectos farmacológicos de los bloqueantes neuromusculares; situación que debe ser evitada debido a que condiciona el aumento de la morbimortalidad del paciente. Los resultados sugieren como factores predisponentes de relajación muscular residual, variables como el género y la duración de los procedimientos (que no han sido reportadas en la literatura), entre otras.

Palabras clave: algoritmo genético, testores típicos, relajación residual neuromuscular, factores predisponentes.

1. Introducción

La relajación residual neuromuscular (RRNM) es una condición clínica, determinada por la persistencia de los efectos farmacológicos de los bloqueantes neuromusculares; que puede aumentar la morbilidad postoperatoria con incidencia variable entre 0% y 93% [16]. El propósito de este estudio, es la identificación de sus factores predisponentes mediante el uso de una herramienta genética híbrida.

La determinación del conjunto de características que mejor describe un fenómeno, es un problema bien conocido y abordado mediante diferentes tipos de herramientas. Algunas extraídas de la minería de datos [15] y otras, debido a su conocido nivel de complejidad, inspiradas en técnicas no exhaustivas como es el caso de metaheurísticas tanto mono-objetivo [19, 22, 23, 24], como multiobjetivo [13]. La determinación de un subconjunto de características tiene un espacio de búsqueda de 2^n posibles subconjuntos, por lo que se ha catalogado como un problema NP-duro. Para una revisión detallada sobre los marcos de trabajo para éste problema, el lector puede remitirse a [6].

Es importante hacer mención de que para nuestra aplicación, una característica equivale a una variable descriptora de la condición médica del paciente, por tal motivo, en el resto del documento haremos uso indistinto de los términos variable y característica.

El algoritmo que se empleó en este trabajo, está definido dentro de un marco de trabajo adaptable tanto para aprendizaje supervisado como no supervisado [23], además de la inclusión de los operadores de mejora y de aceleración. El algoritmo es conducido mediante la valoración desde una perspectiva de testor de los individuos generados, consiguiendo una reducción considerable del fenómeno en descripción.

El algoritmo presentado en este documento se basa en el conocido como “algoritmo genético simple” de Goldberg [8], cuyas bases aparecieron a mediados de los años setentas entre Holland y sus colaboradores [9]. La idea central de esta metaheurística está inspirada en la supervivencia de los individuos más aptos, de acuerdo a la teoría de la evolución.

Este trabajo se ha organizado de la siguiente manera: en la sección 2 se discuten algunos conceptos básicos relacionados tanto con la técnica empleada para la solución del problema, como con el problema de relajación residual neuromuscular. En la sección 3, se presenta la descripción del algoritmo empleado para la identificación de los factores predisponentes de RRNM, así como la explicación del operador de aceleración. Los resultados obtenidos son descritos y discutidos en la sección 4, para finalmente mostrar las conclusiones y trabajo futuro en la sección 5.

2. Conceptos básicos

2.1. Testores y testores típicos

El concepto de testor aparece a mediados de los años cincuenta (1950's) [3]. Al principio, los testores típicos se usaron como una herramienta muy eficaz para detectar fallas en circuitos eléctricos. Posteriormente, se usaron para llevar a cabo clasificación supervisada y selección de variables en problemas de geología [1]. El uso que se les da en este trabajo a los testores y testores típicos está estrechamente relacionado con selección de características, cuyo precursor es Dmitriev, Zhuravlev y colaboradores [5].

En esencia, un testor es un subconjunto de características que distinguen objetos provenientes de clases distintas. De acuerdo con Santiesteban y Pons en 2003 [18], así como Shulcloper en 1995 [20], un testor típico es un testor al que ya no se le puede

quitar ninguna de sus características sin que pierda su condición de testor. Dicho de otra manera, un testor es todavía redundante en términos de que aún puede ser un conjunto más pequeño de características pudiendo identificar la clase a la que pertenece el objeto en cuestión; mientras que un testor típico está ya formado por el conjunto de características mínimo necesario para garantizar aún la identificación de la clase a la que pertenece un objeto específico.

Supongamos que U es una colección de objetos, y esos objetos son descritos por un conjunto n de características, los objetos, se encuentran agrupados en k clases. Comparando cada par de características que pertenecen a objetos en clases diferentes haciendo uso de algún criterio de comparación, podemos obtener la **Matriz de diferencias (MD)**. Esta matriz de diferencias puede ser muy grande cuando son encontradas muchas diferencias entre los miembros de una clase contra los de las demás.

Llamémosle T a un subconjunto del total de etiquetas de columna de **MD**. Llamaremos **Matriz Básica (MB)** a un conjunto de datos especial obtenido de la eliminación de todas las filas pertenecientes a la **MD** que no son básicas.

Supóngase que a y b son dos filas de la matriz de diferencias, a es una subfila de b si $(\forall_i | b_i = 0, a_i = 0)$ y además las filas cumplen al menos en una de sus columnas $(i | b_i = 1 \wedge a_i = 0)$ [11]. En otras palabras, a es una fila básica de **MD** si no hay ninguna otra fila menor que a en **MD**.

Dada una Matriz de diferencias **MD**, llamaremos Matriz Básica (**MB**) a la matriz formada exclusivamente por filas básicas de **MD** [11].

T es un testor de la **MA** (Matriz de aprendizaje) si no existen filas de ceros en la **MB** al eliminar de ésta todas las columnas que no pertenezcan al conjunto T .

El conjunto T es típico si al eliminar cualquiera de sus características $j | j \in T$, T deja de ser testor.

El conjunto de todos los testores típicos de la **MD** es igual al conjunto de todos los testores típicos de la **MB** [11].

Calcular todos los testores típicos es un procedimiento muy costoso; de hecho, todos los algoritmos exhaustivos conocidos tienen una complejidad exponencial e incluso dependen del tamaño de la matriz [17].

El lector interesado podrá revisar algunos de los algoritmos desarrollados para el cálculo del conjunto completo de testores típicos como: **BT** [21], **TB** [21], **CT-EXT** [17], **LEX** [18] y **REC** [20] entre otros.

En este trabajo, utilizamos un algoritmo genético que usa testores típicos para encontrar las variables más importantes en relajación residual, así como su peso relativo en términos de su capacidad predictiva. Algunos conceptos e ideas relacionados con la teoría de testores típicos puede ser consultada por el lector en [12].

2.2. Relajación residual neuromuscular

La necesidad de incorporar nuevas guías o estrategias en la buena práctica de uso de los bloqueantes neuromusculares es una necesidad imperiosa en Anestesiología. En la práctica actual sólo existen recomendaciones de actuación con el propósito de convencer de que la monitorización neuromuscular es una herramienta muy útil para el buen uso de los bloqueadores neuromusculares. El único método satisfactorio para

determinar un grado de bloqueo neuromuscular es estimular un nervio motor con una corriente eléctrica y observar el grado de contracción de los músculos inervados por ese nervio. Es fundamental encontrar alternativas que permitan prever problemas en pacientes con predisposición a la relajación residual neuromuscular.

La razón por la que la monitorización neuromuscular no ha recibido la aceptación necesaria en la práctica clínica es un reflejo de la discordancia entre lo que la literatura recomienda y lo que los clínicos pueden medir. En la actualidad, la monitorización de la función neuromuscular, no es una práctica común porque los clínicos no saben interpretar correctamente sus resultados [2]. El uso de un neuroestimulador es más una excepción que una regla en cualquier servicio de anestesia en nuestro país [2]. Podemos afirmar que la relajación residual neuromuscular (RRNM) es una lección que no ha sido asimilada en su importancia en las instituciones de salud mexicanas, es un tópico al que se le menosprecia y al que no se ha dado el valor que merece [14].

A pesar de la alta incidencia de relajación residual neuromuscular (RRNM) en la unidad de cuidados postanestésicos (UCPA), en algunos estudios se reporta de hasta 24% [14], el uso del monitoreo de la función neuromuscular ya sea en cirugía o en recuperación es infrecuente en demasía.

2.3. Monitorización / estimulación

La posible toxicidad de los bloqueadores neuromusculares es alta ya que interfieren con la función del sistema respiratorio.

La depresión ventilatoria es una causa importante de mortalidad y morbilidad relacionadas con la anestesia [14], y no hay duda de que el bloqueo neuromuscular residual en estas circunstancias es un factor importante. Además, los límites de seguridad son reducidos pues el bloqueo se produce en un margen estrecho de ocupación de receptores. Es por eso que es importante que el clínico valore el efecto de los bloqueadores neuromusculares sin la influencia distractora de los agentes volátiles, anestésicos intravenosos y opioides. [7]

El monitoreo de la función neuromuscular tiene las siguientes utilidades:

- a) Administración de la dosis óptima individual de relajantes muscular y antídotos
- b) Administración de estos medicamentos en el momento adecuado e identificar el momento de reversión
- c) Identificación del tipo de relajante en caso de bloqueo residual. Por tanto, se evita la sobredosis, se previene el riesgo de subdosificación – que puede ser de graves consecuencias en neurocirugía o en cirugía de cámara anterior del globo ocular abierta, entre otras, permite administrar el anticolinesterásico en el momento adecuado, se utiliza menor cantidad de medicamento, se evita el riesgo de relajación prolongada, y en consecuencia la recuperación es más rápida. Una alternativa importante es conocer con antelación las características del paciente que le predisponen fisiológicamente a manifestar una relajación residual neuromuscular para tener precauciones especiales con esos casos aun cuando no se cuente con el neuroestimulador (que es un equipo de monitorización costoso).

2.4. Selección de características

Selección de Subconjuntos de Características (FSS por sus siglas en inglés) es un tópico frecuentemente usado para reducir dimensionalidad. En este sentido existen dos grupos de autores que pueden ser ampliamente identificados: aquellos quienes consideran la importancia o relevancia de las variables, así como la no redundancia de las mismas [4] y aquellos que consideran la importancia de distinguir entre las características óptimas y las relevantes [10]. Los dos métodos más populares para realizar selección de características son los métodos Wrapper y Filter [15], [10].

3. Algoritmo utilizado

3.1 Marco de trabajo

El marco de trabajo de la presente investigación se muestra en el diagrama de la Fig. 1.

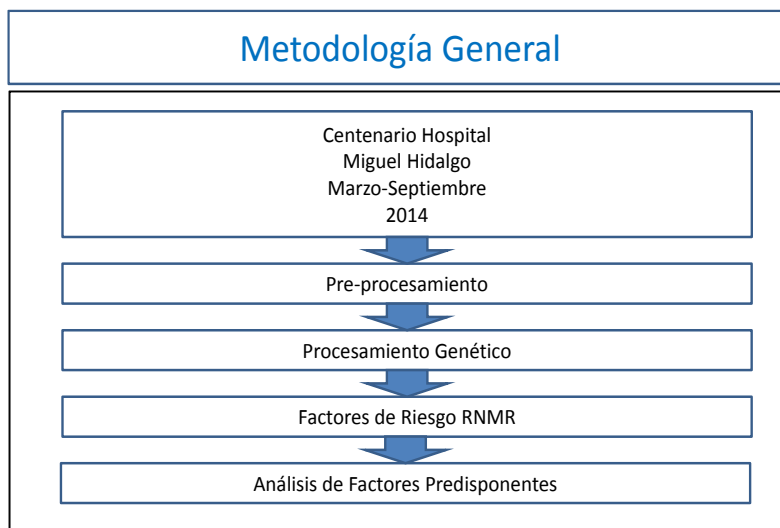


Fig. 1. Marco de trabajo

Como se puede ver en la Fig. 1, se colectó la información de todos los casos que fueron autorizados para ser monitorizados en el turno matutino de Marzo a Septiembre de 2014 en el Centenario Hospital Miguel Hidalgo de la ciudad de Aguascalientes, México. Con esta información, se realizó un pre-procesamiento que permitió reducir el conjunto de variables de 25 a 16. Posteriormente, se realizó el procesamiento genético descrito en la sección 3.3 y se redujo el conjunto a 10 variables que fueron analizadas de manera más profunda. Y finalmente, el personal médico realiza una valoración y análisis de los resultados del algoritmo ejecutado.

3.2 Operador de aceleración y mecanismo de mejora

Es un operador de aceleración que permite que los individuos que se van generando de manera aleatoria, se manipulen para que se mantengan en el espacio de soluciones de testores, ahorrando recursos computacionales.

Este operador fue desarrollado por Torres (2010) [23] y permite hacer un análisis simple de la matriz básica para dirigir la creación de individuos hacia un espacio de soluciones con cierta calidad garantizada.

El mecanismo de mejora se presenta en (Torres, 2010) [23] y consiste en una mutación inteligente, que agrega o elimina bits de para encontrar el testor típico dentro de un testor (que no es típico) o para encontrar el testor partiendo de una cadena que no califica como testor.

3.3 Pseudocódigo

El siguiente pseudocódigo representa el comportamiento general del algoritmo Evolutivo Híbrido realizado para la identificación de las principales variables que predisponen a la presentación de relajación residual neuromuscular.

```
Begin /* Algoritmo Evolutivo Híbrido */      Algoritmo - AGERRNM
begin /* Algoritmo Genético Evolutivo para Relajación residual
neuromuscular */
  Pre-procesamiento inicial
  Fase de generación de la MD.
  Fase de generación de la MB.
  Generación de la población inicial (Aleatoriamente).
  Aplicación del operador de aceleración.
  Aplicación del mecanismo de mejora.
  Cálculo del fitness
  Nueva Población <- Población Inicial
  Repeat
    Begin /* Nueva generación con la Nueva Población*/
      Repeat
        Begin /* Ciclo reproductivo por pares */
          Aplicación del operador de selección.
          Aplicación del operador de cruzamiento.
          Aplicación de operador de aceleración.
          Cómputo del fitness.
          Aplicación del mecanismo de mejora.
          Cálculo del fitness.
        End.
      Until Contador = (Tamaño de la generación / 2).
      Ordenamiento de la población por fitness.
      Aplicación de Elitismo.
      Población <- Nueva Población
    End.
```



```

Until (Alcanzar el criterio de parada)
Conjunto final de testores típicos encontrados.
Cómputo del peso informacional de cada característica
Juicio final del conjunto de características escogidas
End /* AGERRNM /*
END /* Algoritmo Evolutivo Híbrido */
    
```

El algoritmo mostrado es un algoritmo genético (AG) en el que se integraron 500 individuos por generación y que fue ejecutado por 300 generaciones usando una probabilidad de cruzamiento alta (80%), cuya mutación es inteligente (mecanismo de mejora) es aplicada en el 100% de los individuos y el operador de aceleración también.

4. Experimentos y resultados

Se evaluaron un total de 210 pacientes que ingresaron a la Unidad de cuidados postanestésicos (UCPA) del Centenario Hospital Miguel Hidalgo (CHMH) que cumplieron con los criterios de selección. 46 pacientes tuvieron relajación residual neuromuscular (RRNM) a su ingreso a la UCPA (21.9%).

Como se mencionó anteriormente, la RRNM es un efecto indeseable en pacientes que salen de un procedimiento quirúrgico porque pueden caer en amnea (insuficiencia respiratoria), si se marean y vomitan, pueden brinco-aspirar, etc.

Esta condición aumenta en general el índice de morbimortalidad pues el paciente no se encuentra completamente recuperados de los relajantes musculares usados en anestesia.

Tabla 1. Peso informacional de las variables arrojadas por el algoritmo

No.	Variable	Peso Informacional
1	Género	100%
2	Anestésico	48%
3	Oximetría	40%
4	Clasificación peso	39%
5	Temperatura	39%
6	Tiempo Anestésico	37%
7	Relajante	36%
8	Monitoreo	36%
9	Clasificación por edad	29%
10	Reversión	29%

Después de ejecutar el algoritmo mostrado en la sección 3.3, pudimos reducir las variables que predisponen la relajación residual neuromuscular de 26 hasta 10 al final

del proceso. La calificación en términos del peso informacional, (o sea el nivel de importancia de las variables para predecir la relajación residual neuromuscular) se muestra en la Tabla 1

Aunque la cantidad de mujeres intervenidas quirúrgicamente en el Centenario Hospital Miguel Hidalgo es mayor que la de hombres, tuvimos un peso informacional de 100% en la variable género como resultado del algoritmo evolutivo basado en testores.

Este resultado generó sorpresa en el personal médico de apoyo a la investigación, sin embargo, al revisar la matriz de entrenamiento de intervenciones quirúrgicas, pudo observarse que en efecto, el porcentaje de las mujeres que tuvieron RRNM (relajación neuromuscular) de 24.82% (35 de 141), es significativamente mayor que el de los hombres 18.64% (11 de 69). Por lo tanto, podemos aceptar que el hecho de ser mujer efectivamente aumenta el riesgo de presentar RRNM. Los médicos pudieron encontrar una explicación práctica a este fenómeno: el tejido adiposo de las mujeres es mayor al de los hombres y los relajantes se absorben en la grasa y se liberan lentamente de este tipo de tejido.

El algoritmo evolutivo arrojó un peso informacional del 29% para la variable edad. Por lo tanto se realizó un análisis de estos resultados y se pudo observar que de los siete pacientes de 12 o menos años analizados en nuestra muestra, ninguno presentó RRNM, esto puede interpretarse como que el riesgo de RRNM aumentará con la edad y disminuye nuevamente en edades avanzadas.

El peso informacional que arrojó la variable ASA (Referente al estado general de salud del paciente) fue de 28%. Al analizar más detenidamente esta variable, observamos que los porcentajes de RRNM fueron 24.39% (20 de 82) para los pacientes ASA I, 18.69% (20 de 107) para ASA II y 28.57% (6 de 21) para ASA III. Este resultado nos inclinaría a asumir que el hecho de tener clasificaciones de ASA III, eleva el riesgo de presentar RRNM.

El índice de masa corporal se clasificó de acuerdo al peso corporal en 4 categorías: Bajo peso, normal, sobrepeso y obesidad.

El peso informacional de la variable IMC (Índice de Masa Corporal) que integra peso y talla del paciente, resultó en 39% por lo que analizamos los resultados de la siguiente manera: El porcentaje de RRNM en pacientes de peso bajo es de 37.5% (3 de 8); el de pacientes con peso normal es de 28.75% (23 de 80); para los pacientes con sobrepeso el porcentaje es de 14.66% (11 de 75); por último para los pacientes con obesidad el porcentaje es de 19.1% (9 de 47). Asumimos en base a este resultado, que el riesgo de presentar RRNM tiene una relación inversa con el peso del paciente, probablemente porque a mayor peso corporal, la dosis de relajantes musculares se calcula de acuerdo al peso ideal, de tal forma que los pacientes con peso bajo o normal reciben mayor cantidad en proporción a su peso de relajantes.

En términos de la variable temperatura, el personal médico corroboró los resultados del algoritmo, pues está reportado en la literatura que la hipotermia potencializa el efecto de los relajantes musculares.

El peso informacional de la variable tipo de anestésico utilizado arrojado por el algoritmo fue de 48%. Este peso informacional es alto, aunque el personal médico atribuyó su peso al hecho de que de los 2 pacientes que se manejaron con propofol y

fentanil, uno presentó RRNM (50%) , y el único paciente que se manejó con una combinación de gases anestésicos presentó el evento(100%). En vista de esto, los médicos fueron muy reservados para aseverar que la combinación de gases anestésicos o la técnica de Anestesia endovenosa total son determinantes de riesgo para presentar RRNM.

La variable administración de anestésicos locales transoperatoria tuvo un peso informacional del 28%. El porcentaje de pacientes a los que se les aplicaron anestésicos locales en algún momento de la cirugía y que tuvieron RRNM es de 20.83% (15 de 72) y el de los pacientes que no los recibieron es de 22.46% (31 de 138). A la luz de estos resultados no podemos considerar que la administración de anestésicos locales durante el procedimiento anestésico sea un factor de riesgo para RRNM en esta investigación.

Otra variable de gran interés fue el tipo de relajante muscular utilizado, se puede caer en el error de pensar que el cisatracurio fue el relajante de mayor riesgo para condicionar RRNM, ya que es el más utilizado en el Centenario Hospital Miguel Hidalgo. Al analizar el porcentaje de pacientes manejados con cisatracurio y que tuvieron RRNM es de 18.88% (34 de 180). Pero al analizar el porcentaje de los pacientes manejados con vecuronio observamos que se dispara al 36%(9 de 25). El único paciente manejado con rocuronio tuvo RRNM 100%; y de los tres pacientes manejados con la combinación de succinilcolina y un relajante no despolarizante, el 66.66% (2 de 3) presentaron RRNM. Por último, el único paciente manejado con una combinación de dos relajantes no despolarizantes, no tuvo RRNM. De lo anterior, podemos asumir que el riesgo de tener RRNM se relaciona en mayor medida al uso de relajantes musculares derivados de esteroides y a la combinación de succinilcolina con un despolarizante, pero este fenómeno deberá estudiarse de manera dirigida.

El monitoreo de la relajación muscular transoperatoria protege a los pacientes de sufrir RRNM presentando un peso informacional del 36%.

La reversión farmacológica transoperatoria del efecto del relajante neuromuscular, presentó un peso informacional de 29%. El porcentaje de pacientes que fueron revertidos y que presentaron RRNM es de 32% (8 de 17) contra el de los no revertidos y que también presentaron RRNM 20.5% (38 de 185). Estos resultados son alarmantes, porque la finalidad de revertir farmacológicamente el efecto de los relajantes musculares es precisamente para evitar la RRNM, lo que no se está logrando.

De lo anterior, el hecho de revertir farmacológicamente el efecto del relajante neuromuscular durante el transanestésico, no previene el evento de RRNM a menos que se monitorice objetivamente la relajación.

El peso informacional de la oximetría calculado por el algoritmo genético fue de 40%. El porcentaje de los pacientes que tuvieron registros de oximetría de pulso de 90% y más (no hipoxemia) y que presentaron RRNM fue de 21.27% (40 de 188). El porcentaje de los pacientes con evidencia de hipoxemia (menor de 90%) y con RRNM fue de 27.2% (6 de 22).

El peso informacional de la variable tiempos anestésicos, fue de 37% y su análisis es el siguiente: el porcentaje de los pacientes con tiempos anestésicos de una hora o menos y que presentaron RRNM fue de 35% (7 de 20); el de los pacientes con tiempos anestésicos de 1 a dos horas fue de 24.73% (23 de 93); por último el de los pa-

cientes con tiempos anestésicos de más de 2 horas fue 16.49% (16 de 97). Estos resultados nos orientan a pensar que el riesgo de presentar RRNM será mayor en pacientes que tengan tiempos anestésicos de una hora o menos, ya que regularmente se les aplica una dosis única de relajante para la intubación y el anestesiólogo asume que el efecto de éste ya ha pasado. Los pacientes con tiempo anestésicos mayores, por obvias razones, recibirán dosis subsecuentes del fármaco, lo que obliga al anestesiólogo a administrar con más cuidado las dosis adicionales.

5. Conclusiones y trabajo futuro

La combinación del algoritmo, con el operador de aceleración, el mecanismo de mejora y la función de evaluación basada en testores, fue capaz de resolver eficientemente el problema de la determinación del conjunto completo de los testores típicos de una base de datos; problema para el que los algoritmos exhaustivos conocidos tienen una complejidad exponencial [17].

La identificación de los 219 testores típicos inmersos en los datos, permitió realizar el cálculo del peso informacional de cada una de las variables, con la consecuente discriminación de aquellas cuyo peso informacional se encontraba por debajo del 29%. Este punto de corte fue establecido por el personal médico especializado en anestesiología que formó parte del equipo de trabajo de la presente investigación.

Las variables que se encontraron por encima de este punto de corte fueron 10, lo que representa una reducción del 61.5% para considerarse como principales factores predisponentes en la relajación residual neuromuscular.

Como trabajo futuro, se incluirá una comparación de los resultados de este análisis con técnicas estadísticas clásicas, así como la integración de los mecanismos descritos en un Algoritmo de Estimación de la Distribución. Otro aspecto que se pretende agregar a la herramienta es el establecimiento automático del punto de corte. El equipo de trabajo también seguirá probando las herramientas con otros problemas médicos.

Referencias

1. Alba-Cabrera, E., Santana, R., Ochoa-Rodríguez, A., Lazo-Cortes, M.: Finding Typical Testors by Using an Evolutionary Strategy. In: Proceedings of V Iberoamerican Workshop on Pattern Recognition, Lisbon, Portugal, pp. 267–278 (2000)
2. Bermúdez, L.: Incidencia de la Relajación Residual Neuromuscular en la Unidad de Cuidados Intensivos. Tesis inédita de Especialidad Médica en Anestesiología, Universidad Autónoma de Aguascalientes, Ags, México (2015)
3. Cheguis, I. A., Yablonskii, S. V.: About testors for electrical outlines. *Uspieji Matematicheskij Nauk*, 4(66), pp. 182–184, Moscow (In Russian) (1955)
4. Dash, M., Liu, H.: Feature selection for classification. *Intelligent Data Analysis*, Vol. 1, pp. 131–156 (1997)
5. Dmitriev, A. N., Zhuravlev, Yu. I., Krendeleiev, F. P.: On the mathematical principles of patterns and phenomena classification. *Diskretnyi Analiz*, 7, pp. 3–15, Novosibirsk, Russia (In Russian) (1966)

6. Dy, J. G. and Brodley, C. E.: Feature Selection for Unsupervised Learning. The Journal of Machine Learning Research. Vol 5. Publisher MIT Press. Cambridge, MA, USA. ISSN: 1533-7928. pp. 845–889 (2004)
7. Murphy, S.G.: Residual neuromuscular blockade and critical respiratory events in the postanesthesia care, *Anesthesia and analgesia*, vol. 32, pp. 107–130 (2008)
8. Goldberg D. E.: Genetic Algorithms in Search, Optimization, and Machine Learning, New York. Addison-Wesley (1989)
9. Holland J. H.: Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence, Ann Arbor. The University of Michigan Press. 1975. Second edition. Cambridge, MA: The MIT Press (1992)
10. Kohavi R. G. H. Jhon.: Wrappers for Feature Subset Selection. *Artificial Intelligence*, 97, pp 273–324. (1997)
11. Lazo-Cortés M., Ruiz-Shulcloper J.: Determining the feature relevance for non classically described objects and a new algorithm to compute typical fuzzy testors *Pattern Recognition Letters*, vol. 16, pp. 1259–1265 (1995)
12. Lazo-Cortes, M., Ruiz-Shulcloper, J., Alba-Cabrera, E.: An Overview of the Evolution of the Concept of Testor. *Pattern Recognition*, vol. 34, pp. 753–762 (2001)
13. Mierswa, I., Wurst, M.: Information Preserving Multi-Objective Feature Selection for Unsupervised Learning. Proc. of the Genetic and Evolutionary Computation Conference (GECCO 2006), pp. 1545–1552, New York, USA, ACM Press. (2006)
14. Miller E.L. RD: *Anestesia de Anestesia*, Elsevier, pp. 811–821 (2000)
15. Peng L., Jiaxian Z., Lanjuan L., Yanhong L., Xuefeng Z.: Data mining application in prosecution committee for unsupervised learning. *Services Systems and Services Management*, 2005. Proceedings of ICSSSM '05. 2005 International Conference on. Volume 2, 13-15, pp 1061 – 1064 Vol. 2. (2005)
16. Salomé B., Viana C.E., Cohelo V., Silva, A.: Bloqueo Neuromuscular Residual después del Uso del Rocuronio o Cisatracúrio, *Revista Brasileña de Anestesiología*, 55: 6: pp 612–616, (2005)
17. Sanchez-Diaz, G., Lazo-Cortes, M.: CT-EXT: An Algorithm for Computing Typical Testor Set, Springer-CIARP07, pp 506–514, (2007). Proceedings. Lecture Notes in Computer Science 4756 Springer 2008, ISBN 978-3-540-76724-4, (2008)
18. Santiesteban, A. Y., Pons P. A.: LEX: Un Nuevo Algoritmo para el Cálculo de los Testores Típicos, *Revista Ciencias Matemáticas*, Vol.21, No.1, pp. 85–95. ISSN: 0256-5374. (2003)
19. Saeys, Y., Degroeve, S., Van de Peer, Y.: Digging into Acceptor Splice Site Prediction: An Iterative Feature Selection Approach. *PKDD* 386–397 (2004)
20. Shulcloper, J. R., Alba-Cortes, Lazo, C.: Introducción al reconocimiento de Patrones: Enfoque Lógico Combinatorio, Serie Verde, No. 51, CINVESTAV-IPN. México . pp. 188. (1995)
21. Shulcloper, J. R, Bravo M. A., Lazo-Cortes.: Algoritmos BT y TB para el cálculo de todos los test típicos. *Revista Ciencias Matemáticas* Vol. VI (2) pp.11–18. Cuba. (1985)
22. Torres M.D, Ponce E.E, Torres A, Luna F.J.: Selección de Subconjuntos de Características en Aprendizaje no Supervisado utilizando una Estrategia Genética en Combinación con Testores y Testores Típicos. Tercer Congreso Internacional de Computación Evolutiva (COMCEV 2007). ISBN: 9707280557. pp 57–63 (2007)
23. Torres M. D. *Metaheurísticas Híbridas en Selección de Subconjuntos de Características para Aprendizaje no Supervisado*. (Tesis inédita de Doctorado). Universidad Autónoma de Aguascalientes, Aguascalientes, México (2010)

María Dolores Torres Soto, Aurora Torres Soto, María de la Luz Torres Soto, et al.

YongSeog K., W. Nick Street, Menczer, F., Feature selection in unsupervised learning via evolutionary search. KDD: 365-369 (2000)

Metamodelo de aprendizaje estratégico (MAE): Arquitectura de la capa de infraestructura, solución basada en la Cloud Computing

Rafaela Blanca Silva-López, Iris Iddaly Méndez-Gurrola, Oscar Herrera Alcántara

Universidad Autónoma Metropolitana, Distrito Federal,
México

{rbsl,ohc}@correo.azc.uam.mx, iddalym@yahoo.com.mx

Resumen. En este trabajo se presenta el uso de *Cloud Computing* como infraestructura de un sistema de gestión del aprendizaje (LMS). El objetivo principal es que el sistema soporte la atención a cientos de alumnos bajo la modalidad de enseñanza *e-Learning* en una Institución de Educación Superior (IES). Algunas de las arquitecturas reportadas se especializan en otros LMS como Moodle, otras se enfocan en definir capas que integran los recursos software, hardware, los recursos de enseñanza entre otros. Existen trabajos que centran sus esfuerzos en *e-Assessment* basado en la nube. Unos más se enfocan en incluir en su arquitectura la seguridad en SOA para el control de acceso del sistema. La solución presentada en este trabajo involucra la plataforma SAKAI como LMS y la virtualización de servidores para el *Cloud Computing* lo cual provee una arquitectura escalable y de alta disponibilidad, y la cual representa la capa de infraestructura de un metamodelo de aprendizaje estratégico.

Palabras clave: metamodelo de aprendizaje estratégico, *Cloud Computing*, *e-Learning*, *e-Assessment*, sistemas de gestión del aprendizaje.

1. Introducción

El *Cloud Computing* es un modelo que permite compartir un conjunto de recursos computacionales (servidores, almacenamiento, aplicaciones y servicios) en línea a través de redes. Emplea técnicas innovadoras a gran escala para acceder a recursos que se encuentran distribuidos geográficamente. Ofrece tres modalidades de servicios [1]:

- a) Software como un Servicio (Software as a Service, SaaS). El proveedor ofrece aplicaciones que son utilizadas por el usuario final;
- b) Infraestructura como un servicio (Infrastructure as a Service, IaaS). El proveedor proporciona infraestructura (hardware), el usuario debe instalar las herramientas que requiera para desarrollar las aplicaciones que necesita; y
- c) Plataforma como un servicio (Platform as a Service, PaaS). El proveedor ofrece utilerías sobre las que el usuario construye sus aplicaciones [3].

Por otro lado, se tienen cuatro modelos para implementar y ofrecer los servicios en la nube [1]:

- a) Nube pública (Public cloud). El proveedor de servicios en la nube proporciona sus recursos de forma abierta a diversos usuarios;
- b) Nube privada (Private Cloud). El usuario gestiona sus servicios en la nube, manteniendo el control, se centralizan los recursos informáticos lo que brinda flexibilidad en la disponibilidad de los mismos;
- c) Nubes Híbridas (Hybrid Cloud). Este modelo combina servicios que se ofrecen de forma pública y de forma privada; y
- d) Nube Comunitaria (Community Cloud). El proveedor ofrece los servicios que son compartidos en una comunidad cerrada.

La arquitectura del *Cloud Computing* integra las características y los modelos definidos previamente (ver Figura 1).



Fig.1. Arquitectura del *Cloud Computing*.

1.1. Cloud Computing en las IES

Existe una tendencia en el uso de Aplicaciones Orientadas a Servicios (SOA) [2] como mecanismos de infraestructura para la siguiente generación de Sistemas de Aprendizaje que soporten la atención a cientos o miles de alumnos bajo modalidades de enseñanza *e-Learning* [4]. Sin embargo, la eficiencia y rendimiento del hardware suele convertirse en el cuello de botella de estos sistemas. Esto ha provocado la necesidad de tener soluciones con una arquitectura escalable y de alta disponibilidad.

El Cómputo en la nube brinda la posibilidad de tener una arquitectura con servidores en clúster que habilitan la escalabilidad y alta disponibilidad mediante el

uso de servidores virtualizados, además de ofrecer el software como un servicio [5]. Esta solución es una excelente alternativa viable para las Instituciones de Educación Superior como es el caso de la Universidad Autónoma Metropolitana (UAM) [6][7]. Son múltiples los beneficios que se obtienen con el uso del Cloud Computing como infraestructura de los Sistemas de Aprendizaje (LMS), los beneficios integran flexibilidad en la distribución de recursos, así como escalabilidad, almacenamiento en arreglos de discos, poder de cómputo (manejo de multiprocesadores y memoria RAM), lo que redundará en un ahorro económico [8][9].

1.2. Tecnologías de Cloud Computing

Las tecnologías que se encuentran atrás de una implementación de *Cloud Computing* son la virtualización y SOA, mismas que permiten que ésta sea flexible, confiable y extensible. La virtualización posibilita compartir recursos físicos por demanda (memoria, storage, CPU). Al crear una máquina virtual se comparte un sistema operativo y un conjunto de aplicaciones como se muestra en la Figura 2.

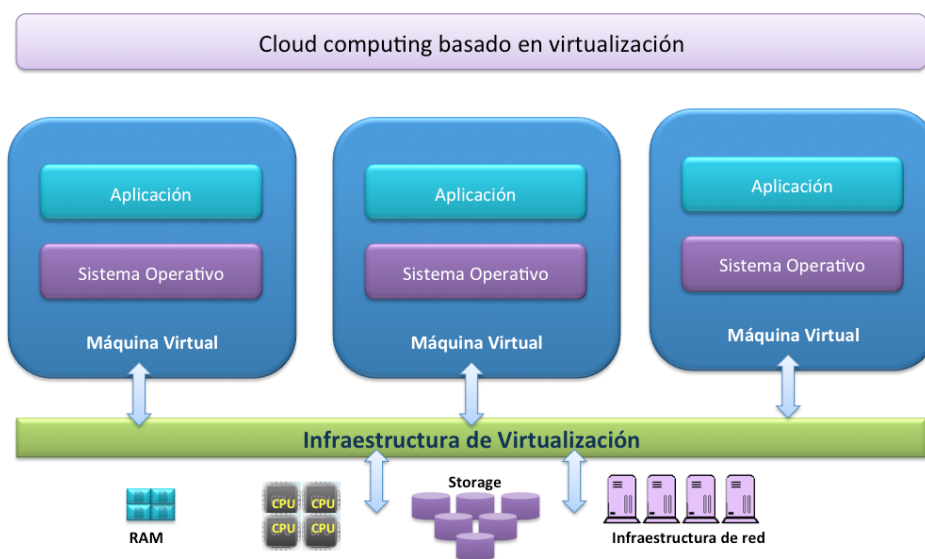


Fig. 2. Cloud Computing con virtualización.

SOA soporta el uso de aplicaciones como servicios para otras aplicaciones. Por lo tanto, es importante el intercambio de datos entre aplicaciones de diferentes proveedores sin programas adicionales.

Web Services

Un *web service* es una aplicación que acepta solicitudes de otros sistemas a través de Internet mediante tecnologías de comunicación independientes de la plataforma. Es

una unidad de software autocontenida, autodescriptiva que proporcionan cierta funcionalidad a través de la infraestructura de la red Internet. El manejo de XML (eXtensible Markup Language) es central a este tipo de arquitectura, por tanto son una herramienta de integración entre plataformas heterogéneas, ya que trabaja típicamente sobre tecnologías abiertas como HTTP (Hypertext Transfer Protocol) y XML¹.

Service Oriented Architecture (SOA)

SOA es un principio arquitectónico para definir sistemas distribuidos empresariales. Es un modelo para la integración de sistemas. Se han definido modelos de referencia como el administrado por OASIS2, que define un vocabulario común para ser utilizado por diferentes organizaciones, busca reducir ambigüedades y problemas de comunicación. Por lo tanto, se implementa utilizando tecnologías como XML, Web Services Description Language (WSDL)³, XML Schema⁴ y SOAP⁵.

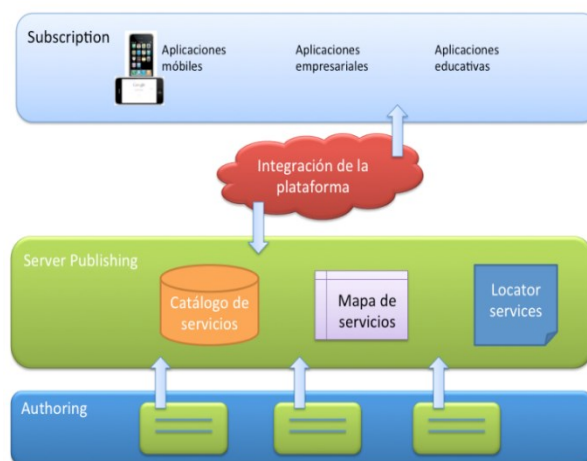


Fig. 3. Implementación de *Cloud Computing* con SOA.

La implementación de *Cloud Computing* aplicando la arquitectura SOA integra el modelo de comunicación de mensajes denominado Publish/Subscribe, en el que los suscriptores son los clientes (aplicaciones móviles, aplicaciones empresariales, entre otras), y los publicadores integran todos los servicios (catálogo de servicios) y ofrecen un mapa de servicios que se tienen disponibles desde múltiples fuentes (ver Figura 3) a través del localizador de servicios.

¹ <http://www.w3.org/XML/>

² http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=soa-rm

³ <http://www.w3.org/TR/wsdl>

⁴ <http://www.w3.org/XML/Schema/>

⁵ <http://www.w3.org/TR/soap/>

2. Arquitecturas para e-Learning basadas en la nube

En esta sección se resumen trabajos asociados con el uso de arquitecturas basadas en *Cloud Computing* para soportar modalidades de enseñanza y aprendizaje mediados por Tecnologías de la Información y Comunicación (TIC). *E-learning* ofrece nuevas alternativas de educación mediante nuevos ambientes virtuales que facilitan la interacción y comunicación entre profesores y alumnos. El uso de la *Cloud Computing* en la arquitectura de Sistemas Basados en Conocimiento (SBC) que apoyan el proceso de enseñanza y aprendizaje es fundamental cuando se pretende atender una gran población de estudiantes que cursan diversos cursos que se apoyan en el uso de TIC.

2.1. Arquitectura de un sistema e-Learning basado en la nube (India)

La arquitectura *e-Learning* propuesta en [10] por los Hindúes, integra cinco capas: *i*) la capa de infraestructura asociada con el hardware *ii*) la capa de recursos de software: sistema operativo y middleware *iii*) la capa de recursos administrativos: servicios por demanda *iv*) la capa de gestión de recursos que permite el acoplamiento flexible del hardware y software, para proporcionar los servicios, y *v*) la capa de servicios considerando SaaS y la capa de aplicación que permite integrar los recursos de enseñanza en nube [10].

2.2. Arquitectura de un sistema e-Assessment basado en la nube (Brasil)

Gusev y otros [11], resumen las cuatro generaciones de sistemas *e-Learning* y *e-Assessment*. La primera generación considera métodos monolíticos basados en tecnologías *stand-alone*; la segunda generación se basa en métodos modulares y tecnologías web; para la tercer generación se aplican métodos orientados a servicios en los que se involucran tecnologías *hypermedia* adaptativa y *web* semántica; finalmente, la cuarta generación usa métodos de cómputo en la nube, soportado por clústers de servidores y la nube. Proponen una arquitectura de *e-Assessment* basada en la nube, como una solución sustentable de bajo costo. La arquitectura **está** constituida por tres módulos: administración, *e-Assessment* y *reports*. El módulo de administración es el núcleo del sistema *e-Assessment*, está constituido por la capa de infraestructura que gestiona el hardware y la capa de servicios que provee los servicios en la nube (*Learning Management System*, *Virtual Learning Environment* y *Management Information System*) [11].

2.3. Smart e-Learning usando Cómputo en la nube y Web 3.0 (Egipto)

Pjanic de la University of Tuzla, Tuzla, Bosnia and Herzegovina, su arquitectura se basa en la solución de cómputo en la nube proporcionada por Microsoft Windows Azure platform y web 3.0, siendo una solución de bajo costo aplicable en Instituciones Educativas [12]. Los componentes de la arquitectura que proponen

incorporar son: a) servicios basados en web (aprendizaje y evaluación); b) modelo basado en una base de conocimiento; y c) el cómputo en la nube utilizando Windows Azure. Los beneficios que se reportan de integrar el cómputo en la nube y la web 3.0 son calidad en el servicio, escalabilidad, uso eficiente de recursos, sustentabilidad, estabilidad, entre otras.

2.4. U-SEA: e-Learning basado en la nube (Brasil)

U-SEA es una propuesta de Piovesan y otros de la Universidad Federal de Santa María (UFSM) de Brasil. Su propuesta se fundamenta en Moodle⁶ y un módulo desarrollado para la adaptación del entorno del estudiante a través de dispositivos móviles, soportado por tecnología de cómputo en la nube. La arquitectura es muy simple, integra como LMS a Moodle y el módulo adaptativo que accede a una base de datos que contiene la información para determinar los mecanismos de adaptación del contexto. Mientras que la infraestructura considera el mecanismo de cómputo en la nube privado, y el modelo de SaaS [13].

2.5. “Cloud Education” (Egipto)

La arquitectura de “Cloud Education” basada en la nube, se constituye a partir de una capa física en la que incluyen el hardware necesario (almacenamiento, servidores, red), una capa de virtualización (servidores, almacenamiento y base de datos virtuales), la capa de middleware educación (plataforma educativa, administrador de profesores, recursos educativos), una capa de aplicación (gestión de usuarios, recursos y cursos). A diferencia de las anteriores arquitecturas considera una arquitectura de seguridad basada en SOA para el control de acceso [14].

2.6. E-Learning en la nube, basado en la especificación IMS-LD. University of Hassan

El modelo propuesto por denominado Instructional Management Systems-Learning Design (IMS-LD), adopta el uso de contenidos, habilita la interoperabilidad, facilita la adaptación de recursos para la enseñanza y el aprendizaje a través de la nube. En su propuesta integran un servicio pedagógico y en su implementación integran el e-Learning y el cómputo en la nube. También contemplan un proceso de diseño instruccional [15].

La arquitectura en la nube para la Universidad contempla un modelo de desarrollo (Nubes públicas, privadas, híbridas y comunitarias), un modelo de servicios (SaaS, PaaS, IaaS) y recursos móviles (*e-Learning*, archivo digital, portales, servicios de *hosting*). Para el caso que presentan, utilizaron Nube comunitaria y PaaS [3].

⁶ <https://moodle.org/?lang=es>

3. Arquitectura de la capa de infraestructura como una solución basada en la nube

En esta sección se propone la arquitectura y organización de la capa de infraestructura del Metamodelo de Aprendizaje Estratégico (MAE) [16].

El MAE integra los principios de la evaluación mediadora, la personalización de las actividades de aprendizaje, la supervisión, seguimiento y atención personalizada, así como el trabajo de las comunidades de aprendizaje con el objetivo de brindar el fortalecimiento de habilidades para el trabajo en equipo, el liderazgo y la negociación, entre otros. El MAE optimiza los recursos físicos y humanos de una institución, buscando disminuir los niveles de deserción e incrementar los niveles de aprobación. En la Figura 4 se muestra el metamodelo, en donde se aprecian las capas de infraestructura, inteligente y reactiva.

La solución busca optimizar los costos y los recursos existentes para atender grupos de 50 a 250 alumnos de ingeniería.

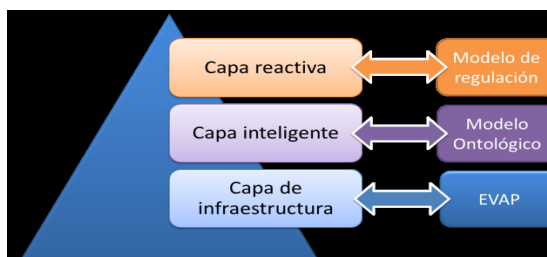


Fig. 4. Modelo de Aprendizaje Estratégico (MAE).

3.1. Arquitectura de la capa de infraestructura del MAE

La arquitectura de la solución de *Cloud Computing* para la capa de infraestructura del MAE, integra cuatro componentes: Learning Management System (LMS), Personalized Virtual Learning Environment (PVLE), Management Information System (MIS) que maneja la información del Ontological Model (OM). Los componentes se implementan como una solución del tipo *Cloud Computing*, tomando como base el trabajo de Christian Gütl (2008) y Gusev, Ristov, Amenski, Velkoski y Bozinovski [11]. Nuestra propuesta integra además el modelo ontológico para la personalización de actividades de aprendizaje que reemplaza el módulo MIS que propone Gusev, como se muestra en la Figura 5.

3.2. Learning Management System (LMS)

Se establece como plataforma LMS a SAKAI⁷, se aprovechan diversas herramientas que se integran como el chat, los foros, los avisos, y se incorpora el uso

⁷ <https://sakaiproject.org/>

de videoconferencia apoyándose en el software de Adobe Open Connect⁸. Adicionalmente se aprovechó el uso de las redes sociales y se determinó la creación de un espacio en *Facebook* para mantener contacto permanente con los alumnos, como se muestra en la Figura 6.

El LMS utilizado es SAKAI, desarrollado en lenguaje Java lo que permite integrar otros desarrollos en el mismo lenguaje, es portable y se puede instalar en múltiples plataformas.

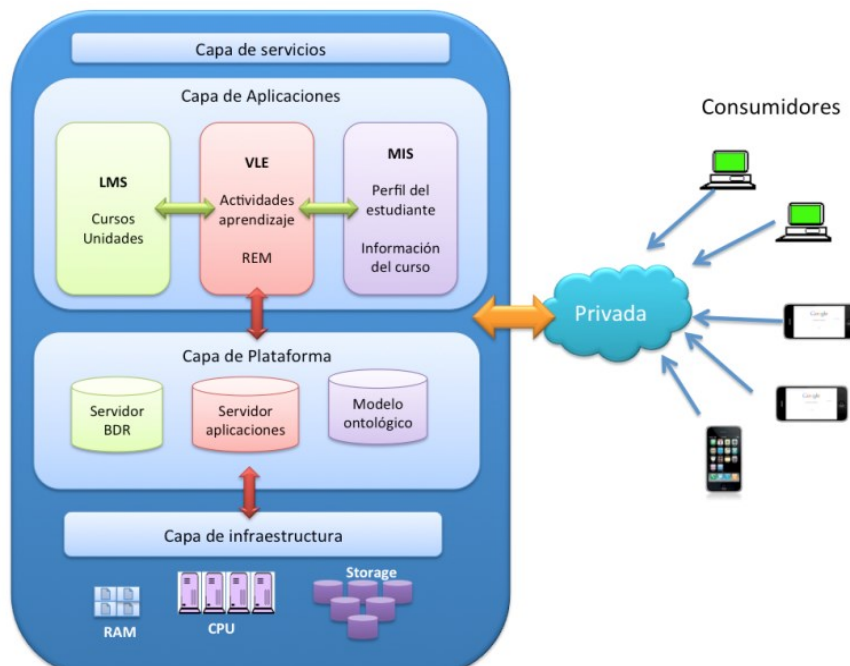


Fig. 5. Arquitectura de integración del Modelo en la *Cloud Computing*.

La capa de datos de SAKAI está desarrollada con el *framework* Hibernate⁹ que permite realizar un mapeo Objeto-Relación (OR) entre los objetos de Java y los campos de las tablas de la base de datos. El mapeo OR habilita la portabilidad para el uso de múltiples manejadores de bases de datos con el simple hecho de cambiar el driver correspondiente. En la Figura 6 se muestra la arquitectura del LMS SAKAI. La plataforma actualmente soporta hasta 8,000 usuarios (600 concurrentes), debido a los recursos de hardware asignados. La propuesta es escalable en el futuro si aumentara la carga mediante la asignación de más recursos de procesamiento, memoria y almacenamiento mediante la infraestructura de virtualización.

⁸ <http://www.adobe.com/mx/products/adobeconnect/apps.html>

⁹ <http://hibernate.org/ogm/>

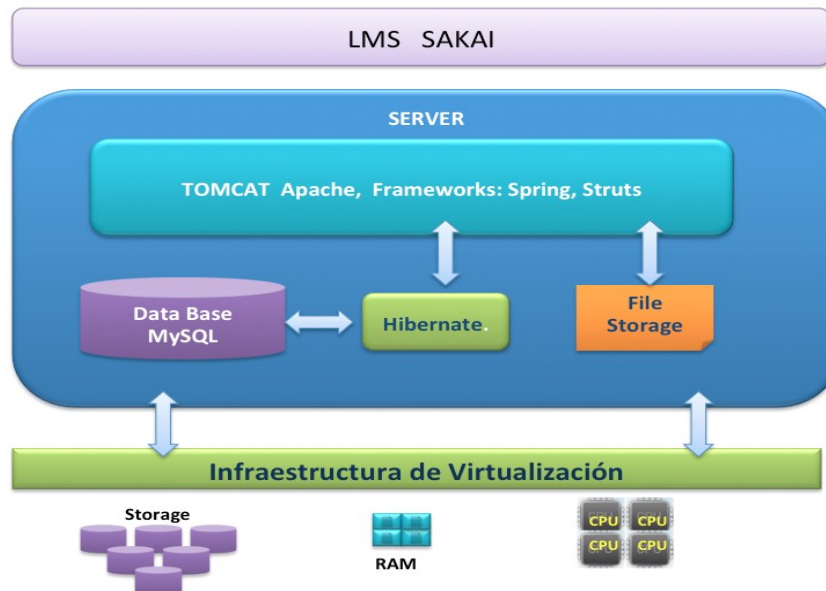


Fig. 6. Arquitectura de soporte para el LMS SAKAI.

3.3. Entorno virtual de aprendizaje personalizado y modelo ontológico

Para llevar a cabo el desarrollo, implementación y pruebas, del Entorno Virtual de Aprendizaje Personalizado (EVAP), es conveniente definir los componentes que se usará como fundamento tecnológico.

El EVAP es un entorno adecuado para el aprendizaje, ya que en él se integran múltiples herramientas que tradicionalmente se usan en Internet tal como el uso de redes sociales, así como recursos educativos multimedia. Para tal fin se elaboraron diversos recursos educativos multimedia, así que el alumno tiene la libertad de usar el recurso que desee. Inclusive, si el tema no le queda claro a partir de un recurso, puede acceder los recursos de otro estilo y complementar su aprendizaje; al mismo tiempo que desarrolla sus otros canales de percepción y habilidades como receptor crítico. Adicionalmente se incorpora el uso y creación de mapas mentales como herramienta para integrar el conocimiento.

El EVAP se propone como un sistema en el que se integrará el Modelo Ontológico (MO) como parte del trabajo futuro. El EVAP se constituye por un servidor de aplicaciones, tres frameworks: uno para la centralización de servicios (Spring¹⁰), otro para el mapeo a base de datos (Hibernate), y un tercero para la implementación del controlador y vista (Struts¹¹). Considera la conexión a ontologías que conforman el modelo ontológico para la personalización de actividades de aprendizaje como se muestra en la Figura 7.

¹⁰ <http://projects.spring.io/spring-framework/>

¹¹ <https://struts.apache.org/>

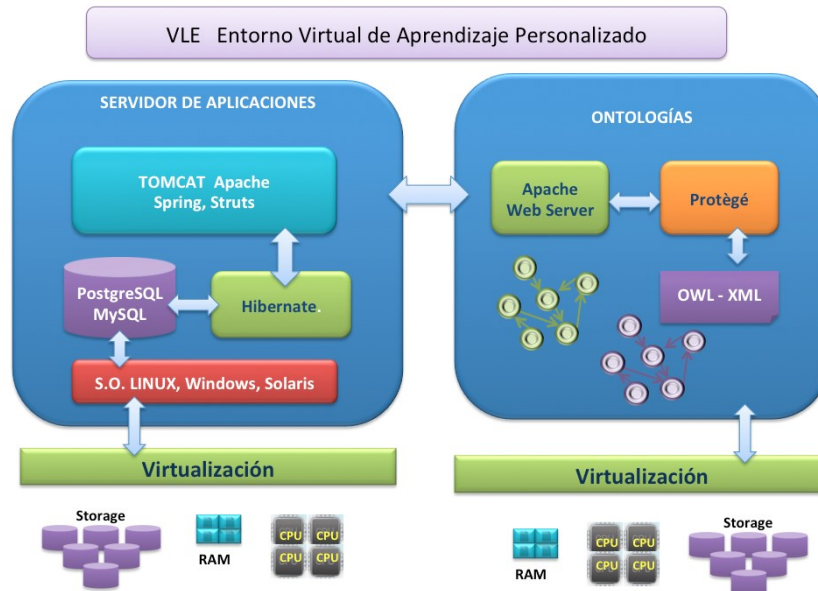


Fig. 7. Arquitectura del EVAP y Modelo ontológico.

3.4. Infraestructura tecnológica

Teniendo en cuenta las bondades que ofrecen las soluciones de *Cloud Computing* se presenta una arquitectura tecnológica que integra los componentes de la solución en la que se incluye el modelo ontológico propuesto en este trabajo de investigación.

La arquitectura utiliza como modalidades de servicios *Plataform as a Service (PaaS)* [3], ya que proporciona el acceso a manejadores de bases de datos (*MySQL*¹², *PostgreSQL*¹³), servidores de aplicaciones (*Apache TOMCAT*¹⁴), entornos de programación (*Eclipse*¹⁵, *NetBeans*¹⁶), compiladores (*C*, *C++*, *Java*¹⁷, *PHP*¹⁸, entre otros), y sistemas operativos (*Linux*¹⁹, *Solaris*²⁰, *Windows*²¹), sobre las que el usuario construye las aplicaciones que necesita. Los productos y marcas son propiedad de las respectivas compañías.

Como modelo de implementación para ofrecer los servicios en la nube se utiliza tanto la *Private Cloud (Nube privada)* como la *Community Cloud (Nube*

¹² <https://www.mysql.com/>

¹³ <http://www.postgresql.org/>

¹⁴ <http://tomcat.apache.org/>

¹⁵ <https://eclipse.org/>

¹⁶ <https://netbeans.org/>

¹⁷ <https://www.java.com/es/>

¹⁸ <https://php.net/>

¹⁹ <https://www.linux.com/>

²⁰ <http://www.oracle.com/us/products/servers-storage/solaris/solaris11/features/index.html>

²¹ <http://windows.microsoft.com/es-mx/windows/home>

Comunitaria). Los servicios son compartidos en una comunidad cerrada al interior de la Universidad.

Por último, la tecnología base para la implementación de *Cloud Computing* es la virtualización, ya que ofrece disponibilidad, flexibilidad, confiabilidad y extensibilidad. Los recursos de espacio en disco, memoria y CPU se distribuyen dinámicamente de acuerdo con la demanda en las máquinas virtualizadas disponibles, como se muestra en la Figura 7.

La infraestructura tecnológica con la que cuenta la Institución de Educación Superior incluye 4 servidores *Blade HP Proliant BL 465c G7*, cada uno con 24 microprocesadores *AMD Opteron 6174* de 2.199 GHz, con licencia *VMware vSphere 5 Desktop* y 16 GB de memoria RAM configurada por servidor, ver Figuras 8 y 9.

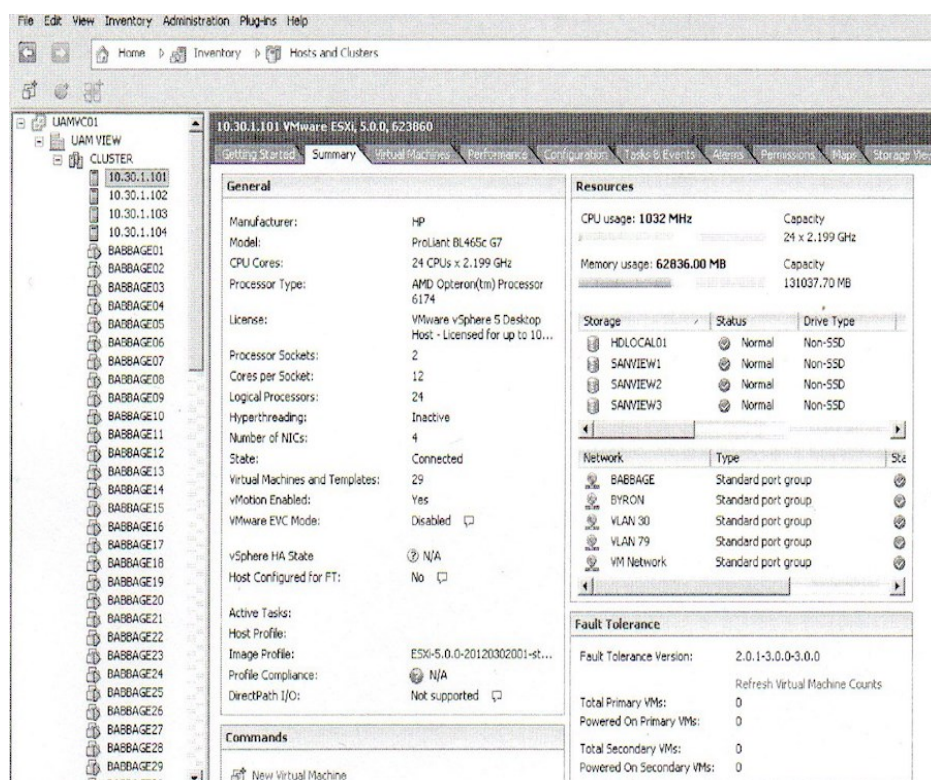


Fig. 8. Servidores Blade HP Proliant BL 465c G7, con VMware vSphere 5.

Se utiliza el software *VMware vSphere 5* para llevar a cabo la virtualización de los recursos que se distribuyen de manera dinámica con el objetivo de optimizar los recursos existentes.

En cuanto a la seguridad, se aplica el aislamiento con el hipervisor incorporado, detección de intrusiones, antivirus, escaneo de vulnerabilidad. Se aplican los mecanismos de seguridad específicos de las VM incrustadas en APIs del hipervisor

utilizadas para proporcionar seguimiento granular del tráfico. Se tiene un política de respaldos y restauración que contempla la gestión de cambio y configuración, considerando la gestión de parches, mantenimientos programados trimestralmente.

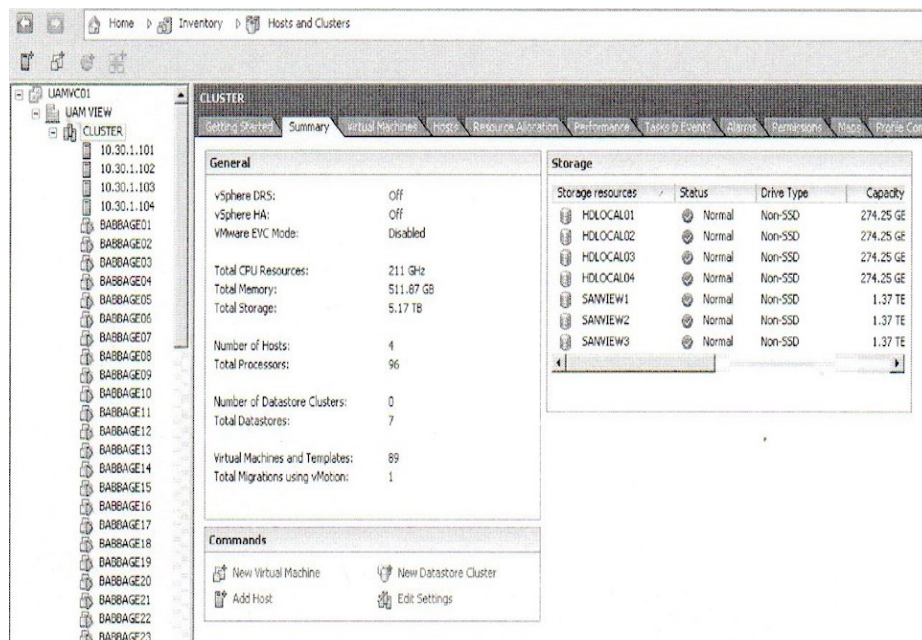


Fig. 9. Cluster de Servidores Blade HP Proliant BL 465c G7, con VMware vSphere 5.

4. Conclusiones

En este trabajo, se describe la arquitectura de la capa de infraestructura del metamodelo de aprendizaje estratégico (MAE), implementado como una solución en la nube. Uno de los puntos importantes de nuestra arquitectura a diferencia de las presentadas en la sección 2, es la incorporación de las arquitecturas de los cuatro componentes del MAE: LMS, el EVAP, el Management Information System (MIS) y el MO. La implementación en la nube se realiza a partir de servidores virtualizados que utilizan recursos físicos de hardware sobre demanda. El modelo de implementación utilizado es tanto Private Cloud, como Community Cloud ya que los servicios se comparten al interior del Campus de la Institución de Educación Superior. En cuanto a la modalidad del servicio se utiliza PaaS para tener acceso a servicios como el manejador de la base de datos, los servidores de aplicaciones y los entornos de programación. La seguridad es revisada a través del monitoreo de las aplicaciones y las bases de datos que se integran en el modelo propuesto. Dado que se implementó un esquema PaaS, en una Cloud Private, los Sistemas Operativos virtualizados incluyen controles de seguridad por capas: a nivel aplicación (con controles de

acceso), a nivel base de datos, a nivel red (firewall). Se crearon zonas de seguridad y se dividieron los ambientes de desarrollo y producción.

Como trabajo futuro se espera integrar ontologías asociadas con los estilos de aprendizaje y pensamiento que faciliten a su vez la integración de técnicas de inteligencia artificial para personalizar la enseñanza y el aprendizaje del alumno.

Referencias

1. Machado, J. C.; Moreira, L. O., Sousa, F. R. C.: *Computação em Nuvem: Conceitos, Tecnologias, Aplicações e Desafios*. In: ERCEMAPI (2010)
2. Blinco, K., Mason, J., McLean, N., Wilson, S.: *Trends and issues in e-learning infrastructure development*. Altilab04, Redwood City, California, USA (2004)
3. Jansen, W. A.: *Cloud Hooks: Security and Privacy Issues in Cloud Computing* NIST Proceedings of the 44th Hawaii International Conference on System Sciences (2011)
4. Dagger, D., O'Connor, A., Lawless, S., Walsh, E., Wade, V.: *Service oriented e-learning platforms: From monolithic systems to flexible services*. *Internet Computing, IEEE*, vol. 11, no. 3, pp. 28–35, <http://dx.doi.org/10.1109/MIC.2007.70> (2007)
5. Armbrust, M., Fox, A., Griffith, R., Joseph, A., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I. et al.: *A view of cloud computing*. *Communications of the ACM*, vol. 53, no. 4, pp. 50–58, <http://dx.doi.org/10.1145/1721654.1721672> (2010)
6. Mircea, M., Andreescu, A.: *Using cloud computing in higher education: A strategy to improve agility in the current financial crisis*. *Communications of the IBIMA*, vol. 2011 (2011)
7. Chandra, D., Borah, M.: *Cost benefit analysis of cloud computing in education*. In: *Computing, Communication and Applications (ICCCA), 2012 International Conference on, IEEE*, pp. 1–6 (2012)
8. Ghazizadeh, A.: *Cloud computing benefits and architecture in elearning*. In: *Wireless, Mobile and Ubiquitous Technology in Education (WMUTE), 2012 IEEE Seventh International Conference on*, pp. 199–201 (2012)
9. Fernández, A., Peralta, D., Herrera, F., Benitez, J.: *An overview of e-learning in cloud computing*. In: *W. on Learning Techn. for Educ. in Cloud (LTEC'12)*, ser. *Advances in Intelligent Sys. And Comp.*, L. Uden et al., Ed., Springer, vol. 173, pp. 35–46 (2012)
10. Vasanthi, M., Magboul, I., Anandhavalli, M.: *Moving towards Virtual Learning Clouds from Traditional Learning*. *Higher Educational Systems in India*, Vol. 9, No. 9, Special Issue “Blended Learning” (2014)
11. Gusev, M., Ristov, S., Armenski, G., Velkoski, G., Bozinovski, K.: *E-Assessment Cloud Solution: Architecture, Organization and Cost Model*. *Ss. Cyril and Methodius University, Skopje, Macedonia*, Volume 8, Issue 2, <http://dx.doi.org/10.3991/ijet.v8iS2.2783> (2013)
12. Nasr, M., Ouf, S.: *A Proposed Smart E-Learning System Using Cloud Computing Services*. In: *PAAS, IAAS and Web 3.0*, Helwan University, Helwan, Egypt; *iJET*, Volume 7, Issue 3, <http://dx.doi.org/10.3991/ijet.v7i3.2066> (2012)
13. Piovesan S., Amaral E., Arenhardt C., and Medina R.: *U-SEA: A Learning Environment Ubiquitous Using Cloud Computing*. *Universidade Federal de Santa Maria – UFSM, Santa Maria, Brazil*; *iJET*, Volume 7, Issue 1, <http://dx.doi.org/10.3991/ijet.v7i1.1838> (2012)
14. Hosam, F., El-Sofany, A. A., Khalid, A., Samir, A.: *The Impact of Cloud Computing Technologies in E-learning*. *British University in Egypt (BUE), Cairo, Egypt*; *iJET*, Volume 8, Issue 1, <http://dx.doi.org/10.3991/ijet.v8iS1.2344> (2013)

15. Ouahabi, S., Eddaoui, A., Labriji, E.H., Benlahmar, E., Guemmat, K.: Implementation of a Novel Educational Modeling Approach for Cloud Computing. University of Hassan II, Mohammedia, Casablanca, Morocco; iJET, Volume 9, Issue 6, <http://dx.doi.org/10.3991/ijet.v9i6.4153> (2014)
16. Silva-López, R., Méndez-Gurrola, I., Sánchez-Arias, V.: Strategic learning, towards a teaching reengineering. *Research in Computing Science*, vol. 65, pp. 133–145 (2013)

Reviewing Committee

Ricardo Acevedo	Ivan Adrian Lopez Sanchez
Moisés Alencastre Miranda	Antonio Marin-Hernandez
Roberto Alonso Rodriguez	Lourdes Martínez
Joanna Alvarado-Uribe	Martinez Medina
Gustavo Arroyo Figueroa	Miguel Angel Medina Perez
Christian Arzate	Efrén Mezura-Montes
Ivonne Maricela Ávila Mora	Sabino Miranda-Jiménez
Jorge Bautista López	Daniela Moctezuma
Ivo Buzon	Raul Monroy
Maria Victoria Carreras	Jaime Mora-Vargas
Felix Castro Espinoza	Saturnino Job Morales Escobar
Noé Alejandro Castro Sánchez	Lourdes Muñoz Gómez
Bárbara Cervantes	Antonio Neme
Jair Cervantes	Alberto Oliart Ros
Efren Chavez Ochoa	Mauricio Osorio
Gustavo Delgado Reyes	Elvia Palacios
Sofía N. Galicia-Haro	Hiram Eredin Ponce Espinosa
Natalia Garcia	Carlos Pérez Leguizamo
Alexander Gelbukh	Maricela Quintana López
David Gonzalez	Carlos A. Reyes-García
Miguel Gonzalez-Mendoza	Carlos Alberto Rojas Hernández
Hugo Gustavo González	Rafael Rojas Hernández
Hernández Mario Graff	Dafne Rosso
Fernando Gudiño	Oscar S. Siordia
Pedro Guevara López	Grigori Sidorov
Yasmin Hernandez	Abraham Sánchez López
Neil Hernandez Gress	Israel Tabarez Paz
Oscar Herrera	Eric Sadit Tellez
Rodolfo Ibarra	Nestor Velasco-Bermeo
Yulia Ledeneva	Elio Villaseñor
Asdrubal Lopez Chau	Alisa Zhila
Juan Carlos Lopez Pimentel	

Impreso en los Talleres Gráficos
de la Dirección de Publicaciones
del Instituto Politécnico Nacional
Tresguerras 27, Centro Histórico, México, D.F.
mayo de 2015
Printing 500 / Edición 500 ejemplares

