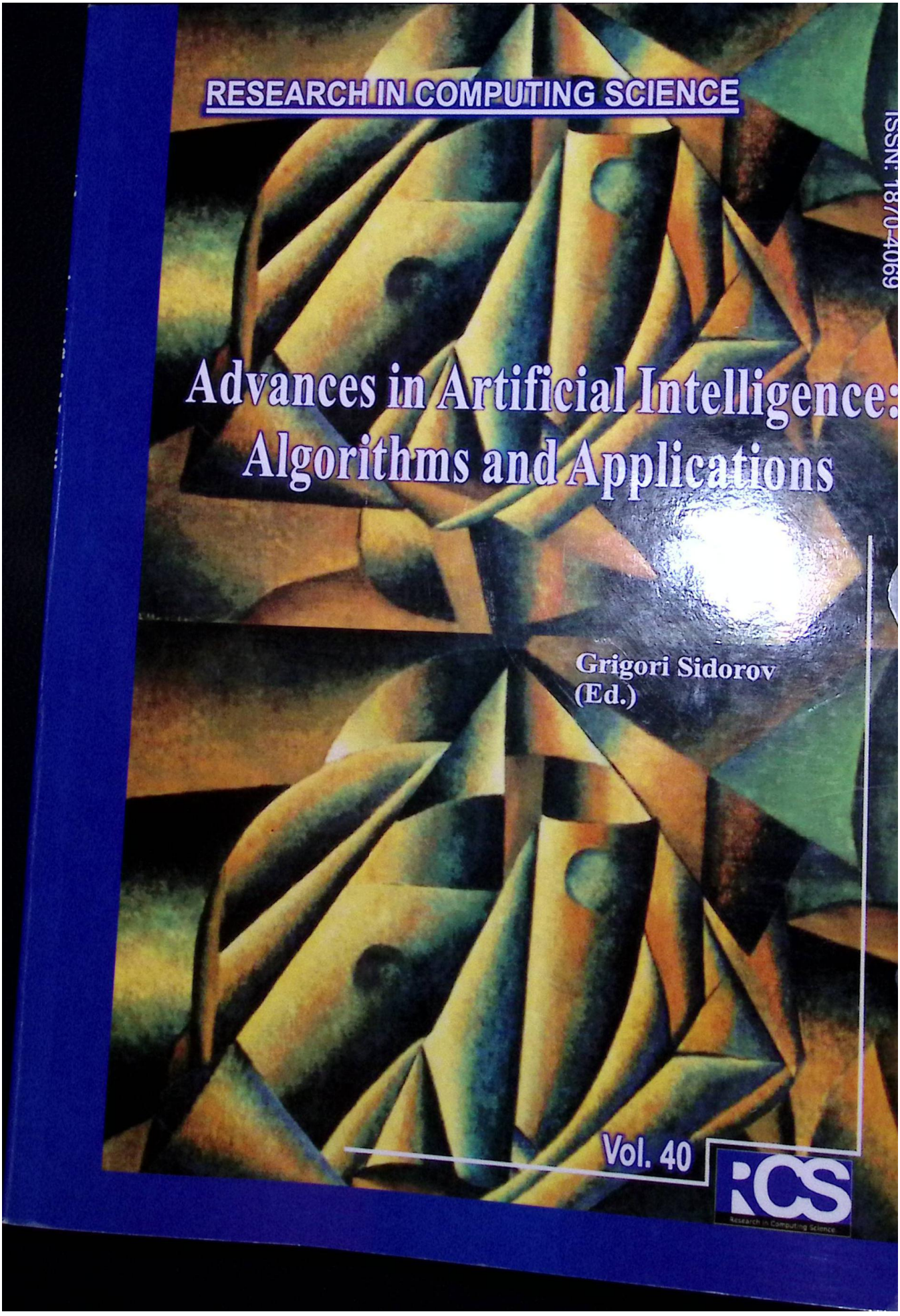

RESEARCH IN COMPUTING SCIENCE
ISSN: 1870-4069
Advances in Artificial Intelligence: Algorithms and Applications
Grigori Sidorov (Ed.)
Vol. 40
RCS
Research in Computing Science

Advances in Artificial Intelligence: Algorithms and Applications

Research in Computing Science

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 40** Octubre, 2008. Tiraje: 500 ejemplares. *Certificado de Reserva de Derechos al Uso Exclusivo del Título* No. 04-2004-062613250000-102, 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. Diseño de la portada se basa en la pintura "Cabeza de una campesina" de Kasimir Malevich. 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: *Juan Humberto Sossa Azuela, RFC SOAJ560723*

***Research in Computing Science** is published by the Center for Computing Research of IPN. **Volume 40**, October, 2008. 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, October, 2008, in the IPN Graphic Workshop – Publication Office.*

Volume 40

Volumen 40

Advances in Artificial Intelligence: Algorithms and Applications

Volume Editor:
Editor de Volumen

Grigori Sidorov

Instituto Politécnico Nacional
Centro de Investigación en Computación
México 2008

ISSN: 1870-4069

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

Indexed in LATINDEX and Periodica / Indexada en LATINDEX y Periodica

Printing: 500 / Tiraje: 500

Printed in Mexico / Impreso en México

Preface

This volume contains contributions on selected topics of artificial intelligence.

The following topics are represented in this volume:

- Self organizing maps,
- Graph matching and pattern recognition,
- Computer security.

In each area, the contributions of this volume are related with development of algorithms and applications.

Self organizing maps (SOM) is a growing are of applications of artificial intelligence. SOM is a special type of artificial neural networks that is usually trained using unsupervised machine learning methods and represents the complex input data as a low dimensional (usually, two-dimensional) picture. This picture corresponds to the "map" of this data. This volume contains several papers on various applications of SOM with wide spectrum of used algorithms: genome analysis, demographic studies, exporting, construction of electoral units, and volcanic domain.

Pattern recognition is a traditional area of artificial intelligence, usually, but not necessarily, related to image processing. It can be viewed as classification problem based on former knowledge obtained from known patterns. At the modern stage, this knowledge is often obtained using statistical or machine learning methods. Graph matching can be viewed as a specific task in pattern recognition, namely, what graphs can be considered similar, and to what extent. The problem of graph matching is notable for its high complexity. The papers in this volume present both graph matching algorithms and pattern recognition algorithms. In the latter case, images are used as the data for processing.

Another important topic of artificial intelligence is computer security. The papers in this volume are dealing with intrusion detection techniques, as well as with more philosophical vision of security as self-healing and self-repairing.

The papers were carefully selected by the international editorial board of the volume, separately for each area, on the basis of thorough reviewing process.

I am sure that these papers will be of interest for the specialists in the corresponding areas, as well as for researchers in other areas of artificial intelligence and for general public interested in the theme.

I would like to express many thanks to Sulema Torres-Ramos for her invaluable help in preparation of this volume.

Grigori Sidorov

Table of Contents

Page

Self-organizing Maps: Algorithms and Applications

Dimer Patterns in Database of Viral Genomes:
An Analysis with GHSOM 3
Ernesto Bautista-Thompson, Gustavo Verduzco-Reyes, and Luis De la Cruz-De la Cruz

The use of Weighted Metric SOM Algorithm as a Visualization Tool
for Demographic Studies 13
Elio Villaseñor, Humberto Carrillo, Nieves Martínez de la Escalera, and Valeria Millán

Determinants of Export Performance:
An Analysis using the SOM Algorithm 27
Omar Neme, Antonio Neme, and Alejandra Cervera

Construction of Autosimilar Electoral Units
using Self-Organizing Maps 39
Alberto García Aguilar, José Carlos Méndez de la Torre, and Leopoldo Trueba Vázquez

A Novel Approach to the Analysis of Volcanic-Domain Data
using Self-Organizing Maps:
A Preliminary Study on the Volcano of Colima 49
JRG Pulido, EMR Michel, MA Aréchiga, and G Reyes

Graph Matching and Pattern Recognition

Shape Decomposition for Graph Representation 63
Bai Xiao and Peter M. Hall

A Critical Examination of Node-Similarity Based
Graph Matching Algorithms 73
Guoxing Zhao, Miltos Petridis, Grigori Sidorov, and Jixin Ma

Thresholding Method based on the Hmax
and Hmin Morphological Operators 83
Edgardo Felipe-Riveron and David Suarez-Hernandez

3-D Fractal Characterization of Tumors
from a Computer Tomography Scan 95
Ernesto Cortés Pérez, Tomás Morales Acoltzi, and Francisco Viveros Jiménez

Computer Security

Self-healing and Self-protecting Computing Systems:
In the Search of Autonomic Computing 109
Luis M. Fernández-Carrasco, Hugo Terashima-Marín, and Manuel Valenzuela-Rendón

Entropy-Based Profiles for Intrusion Detection in LAN Traffic 119
P. Velarde-Alvarado, C. Vargas-Rosales, D. Torres-Román, and A. F. Martínez-Herrera

Intrusion Detection for Mobile Ad-Hoc Networks
based on a Non-Negative Matrix Factorization Method 131
Carlos Mex-Perera, José Zamora-Elizondo, and Raul Monroy

Author Index 141

Editorial Board of the Volume 143

Self-organizing Maps: Algorithms and Applications

(with Jorge Rafael Gutiérrez Pulido)

Dimer Patterns in Database of Viral Genomes: An Analysis with GHSOM

Ernesto Bautista-Thompson[1], Gustavo Verduzco-Reyes[1], and Luis De la Cruz-De la Cruz[2]

[1] Centro de Tecnologías de la Información, DES-DACI, Universidad Autónoma del Carmen, Avenida 56 Número 4,
C.P. 24180 Ciudad del Carmen, Campeche, México
[2]DACB, Universidad Juaréz Autónoma de Tabasco
C. P. 86690 Cunduacán, Tabasco, México
{ebautista, gverduzco}@pampano.unacar.mx, santanadelacruz@gmail.com

Abstract. An analysis about the dimer patterns in a database of 150 genomes of viruses from sixteen taxonomic families was developed with the technique Growing Hierarchical Self-Organized Map (GHSOM), the GHSOM neural network allows the hierarchical clustering of the viruses by their similarity features in our case dimers frequencies. The clusters generated shows certain degree of correspondence with the taxonomic families but a sharp differentiation was not observed. In the case of the Retroviridae family was observed a strong dispersion of their members between different clusters, reflecting diversity in the frequencies of their dimers. Some families are characterized by specific dimers such as: AA, AT, TA, and TT, as the case of the Poxviridae family.

Keywords: Dimer Patterns, Virus Genome, GHSOM.

1 Introduction

Studies of nucleotide sequences from DNA are of interest because the insight that they can provide about the evolutionary processes of species [1]. In particular, the study of dimer sequences is of interest due to the hypothesis that exists a relation between dimer statistical distribution and the basic conditions for DNA physicochemical stability [2, 3], and also because is possible that dimer distribution is related with a genetic signature useful for phylogenetic and taxonomic classification of species based on a underlying level of information not present in trinucleotide sequences (codons) that are known to carry on the coding information in DNA [4].

Different studies are reported in scientific publications about the application of clustering techniques for the analysis of genomic sequences in

Advances in Artificial Intelligence: Algorithms and Applications
Research in Computing Science 40, 2008, pp. 3-12

gene expression studies, comparison of interspecies characteristics, DNA clone classification, analysis of coding and non coding regions in DNA, focused on DNA from bacteria, plants and animals [5, 6, 7, 8], but few studies were found about analysis and search of similarity patterns on viral genomes [4, 9].

In order to search and identified similarity patterns between the viral genomes at the dimer level, we applied the neural network: Growing Hierarchical Self Organized Maps (GHSOM) for the clustering task. We are interested in exploring different machine learning approaches such as GHSOM for the fusion of data from multiple features and origins in order to extract knowledge in genomic databases, and this work is part of such effort.

In section 2, we briefly present the taxonomic information about the database of viruses under study. In section 3, we describe the experimental methodology and the results of the search of dimer patterns with GSHOM. Finally, in section 4 we present the discussion of this work.

2 Taxonomic Features and Database of Viral Genomes

Viruses are one of the most primitive biological forms on earth, although there is a controversy about if they are living forms or not. They are believed to had been components of cells that became autonomous, in fact some virus are similar to portions of DNA sequences of genes, another hypothesis is that viruses evolved from unicellular organism [10]. There are a well known taxonomic classification of viruses based on the type of organization of viral genome, the strategy of viral replication and the structure of the virion [10, 11], but the explosion of taxonomic information available in public data bases thanks to the application of Information Technologies and the sequencing of virus genomes [11, 12], has complicated the analysis of the information and the discovery of new knowledge inside these databases. The application of techniques such as GHSOM for the dual task of clustering and visualization of the results, allows the identification of patterns of interest from the sets of features contained in genomic databases.

The set of viruses under study are representative of different taxonomic families (sixteen families in this study) and sources of different common and non common human and non human diseases [10, 12], see Table 1. We select randomly 150 virus genomes with different features, virus of DNA and RNA, highly aggressive virus as the Zaire Ebolavirus, virus that produce not very dangerous diseases as the Rhinovirus B and Coronavirus. The size of the genomes is also very variable; there are genomes of less than 5,000 bp like the Parvovirus 4 and genomes around 130,000 bp like the Herpesvirus 1 and Herpesvirus 4. All the genomic sequences were taken from the GenBank through the Entrez Documental Retrieval System [11, 13], and loaded inside a database that was built for the management of the collected data and the dimer frequencies data to be generated.

3 GHSOM and Dimer Patterns

Our feature space was the dimers frequencies for each viral genome. The sixteen dimer combinations based on the four bases that form the genomic code are: AA, AC, AG, AT, CA, CC, CG, CT, GA, GC, GG, GT, TA, TC, TG, and TT. In order to be able to compare among different genomes it was necessary to convert the frequencies to percentages, since the genomes are of different size, so each dimer frequency was normalized by the total number of dimers of the corresponding genome. With the frequency data we built an intensity table, in which each row corresponds to a certain virus genome and the columns to the percentage of frequency of each dimer in the genome. This table is then used as input for the GHSOM technique; we apply the GHSOM Toolbox for MatLab® [14] for the generation of the maps. The Table 2 shows examples of the normalized frequency data for some viruses.

Table 1. Examples of different taxonomic features of the viruses under study.

Virus	Family	Molecule	Topology	Capsid	Envelope
Human adenovirus A	Adenoviridae	dsDNA	Linear	Icosahedral	No
Uukuniemi virus	Bunyaviridae	nssRNA	Linear	Helical	Yes
Sapporo virus	Caliciviridae	pssRNA	Linear	Icosahedral	No
SARS coronavirus	Coronaviridae	pssRNA	Linear	Helical	Yes
Sudan ebolavirus	Filoviridae	nssRNA	Linear	Helical	Yes
Dengue virus type 1	Flaviviridae	pssRNA	Linear	Icosahedral	Yes
Hepatitis B virus	Hepadnaviridae	dsDNA	Circular	Icosahedral	Yes
Human herpesvirus 1	Herpesviridae	dsDNA	Linear	Icosahedral	Yes
Measles virus	Paramyxoviridae	nssRNA	Linear	Helical	Yes
Human papillomavirus - 1	Papovaviridae	dsDNA	Circular	Icosahedral	No
Parvovirus H1	Parvoviridae	ssDNA	Linear	Icosahedral	No
Foot-and-mouth disease virus A	Picornaviridae	pssRNA	Linear	Icosahedral	No
Variola virus	Poxviridae	dsDNA	Linear	Icosahedral	Yes
Human immunodeficiency virus 1	Retroviridae	pssRNA	Linear	Helical	Yes
Rubella virus	Togaviridae	pssRNA	Linear	Icosahedral	Yes

Table 2. Examples of normalized frequency values used to build the intensity table.

AA	AC	AG	AT	VIRUS
5.23	6.11	4.79	5.36	HEPATITISB
2.54	6.15	3.99	2.59	HERPES1
4.08	5.89	6.55	3.72	HERPES4
4.54	7.29	4.71	3.81	HERPES5
2.19	5.92	3.91	2.29	HERPES2
9.91	6.15	5.59	7.66	HERPES6
13.24	5.81	5.45	9.47	HERPES7
8.05	6.8	4.7	7.6	HERPES3
5.68	6.88	6.31	4.89	HERPES8
7.71	6.38	6.46	6.02	ADENOB
7.74	6.61	5.66	4.73	ADENOF
5.11	6.5	6.33	4.13	ADENOE
5.88	6.46	6.46	4.55	ADENOD
11.98	5.45	6.3	9.47	HEMORRHAGICENT

The Growing Hierarchical Self Organizing Map (GHSOM) is an unsupervised clustering technique that allows the generation of hierarchies of clusters based on the similarity of the input data, the basis of this map is the SOM neural network that exploits the non supervised competitive learning, the algorithm generates a mapping that preserves the space topology of greater dimension in the space of the neuron units. The neuron units form a two-dimensional grid then a mapping from n-dimension to 2-dimension is generated. The property of topological preservation means that a SOM groups sets of vectors with similar information in neighbor neural units. A SOM network is able to generalize, in this way new information can be added and integrated to the map, also it is able to work with incomplete data inside the vectors [15]. The GHSOM is a variant from the SOM neural network where a hierarchy of multiple layers of SOM neural networks are generated (see Figure 1), each unit of the SOM can generated a new SOM network based on a dissimilarity threshold (quantization error), in this way a hierarchy of similarity clusters is created, the deepness of the hierarchy shows the non uniformity that can be expected from real world data sets [16]. There are other clustering techniques such as K-Nearest Neighbors, Multidimensional Scaling Analysis, Principal Component Analysis, etc. [17], but they are not able to generate a hierarchy of clusters, generalize (preserve the clusters when new information is added), and to work with incomplete input datasets; these features of the GHSOM technique were the factors for its choice as an analytical and visualization tool for the present study.

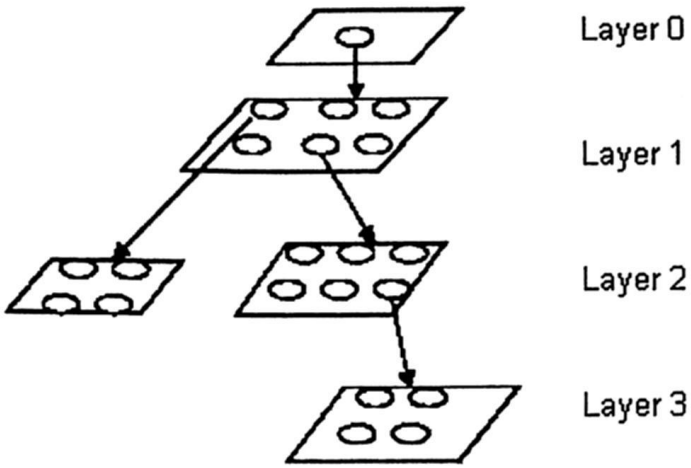


Fig. 1. Hierarchical generation of SOM layers inside a GHSOM neural network, when a new object represented by a node is dissimilar (quantization error) to its neighborhood a new layer is generated through this node, so a new cluster is created.

Instead of associate the specific name of the virus with its corresponding set of dimer frequency data, we associate such data with its corresponding taxonomic family. The generated GHSOM map was analyzed in order to identify global similarities between families of virus; the map shows a similarity hierarchy based on the contributions of the frequency values for the different dimers. Complementary maps were generated that shows in a grey scale the weight of each dimer for different regions inside the GHSOM map. The Table 3 shows the correspondence between the tags in the map, the associated virus family, and the number of virus for each family.

In the GHSOM map (see Figure 2), in general the similarity clusters to which the viruses belongs are in correspondence with the associated taxonomic families (grouping of the different viruses by their corresponding families), but some families presents a strong dispersion of its members: Picornaviridae (tag 13) and Retroviridae (tag 14) families, this shows that differences at the dimer feature level are greater for members of these families, in particular the immunodeficiency viruses belong to the Retroviridae family (see Table 1) and they are known to have a high rate of mutation so their genomic sequences are very variable [10, 18]. Some families have a strong localization of its members: Paramixoviridae (tag 8), Flaviviridae (tag 12) and Togaviridae (tag 15). This is indicative of a strong similarity between the genome of its members at the dimer level.

Table 3. Associated tags for the interpretation of the GHSOM map.

Family	Tag	Number of virus
Adenoviridae	1	7
Hepadnaviridae	2	1
Herpesviridae	3	8
Papovaviridae	4	1
Poxviridae	5	7
Bunyaviridae	6	1
Filoviridae	7	4
Paramyxoviridae	8	18
Rhabdoviridae	9	7
Caliciviridae	10	6
Coronaviridae	11	6
Flaviviridae	12	25
Picornaviridae	13	15
Retroviridae	14	23
Togaviridae	15	15
Parvoviridae	16	6

At the first level of the hierarchy that corresponds to the main four similarity cluster of the GHSOM map, the union of different families can be observed, some examples are in the lower right region of the map: Rhabdoviridae (tag 9), Caliciviridae (tag 10), Flaviviridae (tag 12), Togaviridae (tag 15), and some elements from the Retroviridae family (tag 14). In the lower left region of the map: Poxviridae (tag 5), some elements from the Paramyxoviridae family (tag 8), Coronaviridae (tag 11), and some elements from the Picornaviridae family (tag 13). In the upper left region of the map: Filoviridae (tag 7), Paramyxoviridae (tag 8), and the Parvoviridae family (tag 16). Then, we observed that at the dimer level new similarities between members of different families can be identified with this analysis.

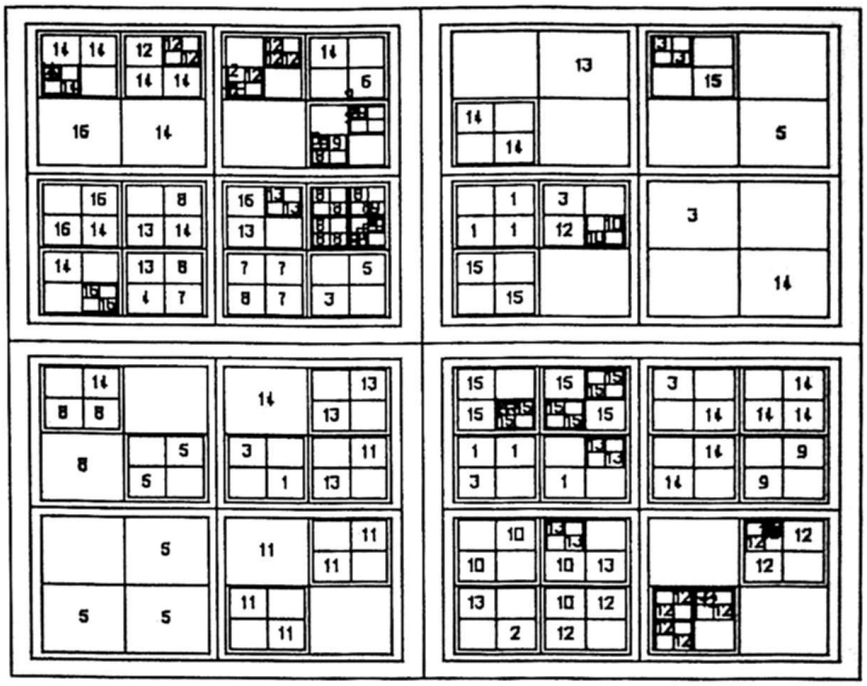


Fig. 2. Hierarchical Map showing the similarity at the taxonomic families level between different viral genomes, where such similarity is based on the dimers frequencies for each genome.

The Figure 3, shows a series of maps that corresponds to the different dimers, each map visualize the weight of a dimer for specific regions inside the GHSOM map, white regions means a strong contribution of the dimer for the elements inside such regions and black regions means a weak contribution of the dimer for the elements inside these regions. The map for the dimer AA shows a white region that corresponds to viruses that belongs to the Poxviridae family, this family is highly localized in the GHSOM map, then the high frequency of occurrence of the dimer characterize to this family. The same case occurs for the dimers AT, TA and TT, so these four dimers have a strong presence in this family of virus.

Another interesting case corresponds to the maps for the dimers: CC, CG, and GC; they show a strong contribution for the viruses grouped on the upper right region inside the GHSOM map.

In the cases of the maps for the dimers: AT, TA, and TT; they have a weak and uniform contribution for the virus grouped in the right side of the GHSOM map, in contrast the maps of the dimers: CC, CG, and GC; show that they have a strong contribution for the same virus grouped in the right side of the GHSOM map.

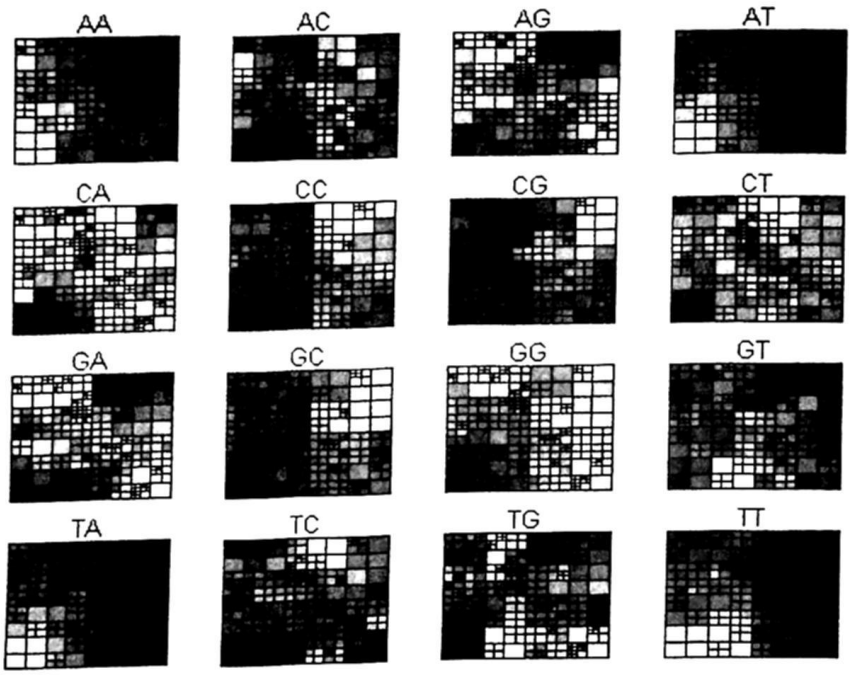


Fig. 3. Maps based on dimer weight, each map shows the contribution of each dimer frequency to the similarity on the corresponding viral genomes associated to each of the regions inside the GHSOM map, the white color corresponds to a strong contribution of the dimer in a specific region and the black color corresponds to a weak contribution of the dimer in a specific region.

4 Discussion

An analysis with the GHSOM technique was developed in order to identify at the dimer frequency level hierarchies of similarity patterns for viruses from different taxonomic families. At the dimer level certain degree of correspondence with taxonomic families was conserved, but a sharp differentiation by families was not observed. An interesting case was the Retroviridae family (which includes the immunodeficiency virus: HIV, SIV, FIV) the members of this family showed a strong dispersion between different clusters reflecting a diversity in the dimer frequency distribution for their genomes. With this analysis was possible to identify similarities between viruses from different families, for example: Filoviridae (tag 7), Paramyxoviridae (tag 8), and the Parvoviridae family (tag 16), where the Filoviridae family has some of the most lethal virus for the human being

(Ebola virus). From the analysis of the dimer contribution to the similarity between the viruses, it was observed that some dimers characterize strongly some families; this was the case of the dimers AA, AT, TA, and TT with the members of the Poxviridae family. The analysis of biological information with clustering techniques such as the GHSOM among others, at the dimer level and its connection with biological knowledge at upper levels such as the taxonomic classification can be a useful tool for the understanding of relations between multiple levels, for example: genomic (1D-structure) and morphological (3D-structure) by combining features from both levels and using techniques such as GHSOM for the identification of patterns of interest. In our case, more research is on the way in order to increment the quantity and detail of the biological data to be integrated from the genomes under study and the exploration and development of techniques for the analysis and visualization of data from genomic databases.

Acknowledgments. The first author thanks the financial support from PROMEP-SEP under the project of generation and application of knowledge UNACAR-PTC-085.

References

1. Stanley, R. H. R., Dokholyan, N. V., Buldyrev, S. V., Havlin, S., Stanley, H. E.: Clustering of Identical Oligomers in Coding and Noncoding DNA Sequences. Journal of Biomolecular Structure & Dynamics 17, 79-87 (1999)
2. Breslauer, K. J., Frank, R., Blöcker, H., Marky, L. A.: Predicting DNA Duplex Stability from the Base Sequence. Proc. Natl. Acad. Sci. USA 83, 3746-3750 (1986)
3. Miramontes, P., Cocho, G.: DNA Dimer Correlations Reflect In Vivo Conditions and Discriminate among Nearest-Neighbor Base Pair Free Energy Parameter Measures. Physica A 321, 577-586 (2003)
4. Quiroz-Gutierrez, A.: Biophysical Considerations and Evolutionary Aspects of DNA-dimer Frequency in AIDS Retrovirus Genomes. In: Topics in Contemporary Physics, pp. 239-248. IPN Press, México (2000)
5. Frith, M. C., Li, M. C., Weng, Z.: Cluster-Buster: Finding Dense Clusters of Motifs in DNA Sequences. Nucleid Acids Research 31, 3666-3668 (2003)
6. Abe, T., Sugawara, H., Kanaya, S., Kinouchi, M., Ikemura, T.: Self Organizing Map (SOM) Unveils and Visualizes Hidden Sequence Characteristics of a Wide Range of Eukariotic Genomes. Gene 365, 27-34 (2006)
7. Figueroa, A., Borneman, J., Jiang, T.: Clustering Binary Fingerprint Vectors with Missing Values for DNA Array Data Analysis. In: Proceedings of the Computational Systems Bioinformatics (CSB'03), pp. 38. IEEE Computer Society Press, U.S.A (2003)
8. McCallum, J., Ganesh, S.: Text Mining of DNA Sequence Homology Searches. Applied Bioinformatics 2(3 Suppl), S59-S63 (2003)
9. Gatherer, D.: Genome Signatures, Self-Organizing Maps and Higher Order Phylogenies: A Parametric Analysis. Evolutionary Bioinformatics Num. 3, 211-236 (2007)

10. Brooks, G. F., Butel, J. S., Ornston, L. N., Jawitz, E., Melnick, J. L., Adelberg, E. A.: Jawitz, Melnick, and Adelberg's Medical Microbiology. Prentice-Hall, U.S.A. (1991)
11. Büchen-Osmond, C.: The Universal Virus Database ICTVDB. Computing in Science & Engineering May/June 2003, 2-11 (2003)
12. Van Regenmortel, M. H. V. et. al. (eds.): Virus Taxonomy. Classification and Nomenclature of Viruses. Academic Press, U.S.A. (2000)
13. The Universal Virus Database ICTVDB, http://www.ncbi.nlm.nih.gov/ICTVdb/
14. GHSOM Toolbox, http://www.ofai.at/~elias.pampalk/ghsom/index.html
15. Kohonen, T.: Self-Organizing Maps. Springer-Verlag, Berlin (2001)
16. Dittenbach, M., Merkl, D., Rauber, A.: The Growing Hierarchical Self-Organizing Map. In: Proceedings of the International Joint Conference on Neural Networks (IJCNN 2000) Vol. 6, pp. 15-19. IEEE Computer Society Press, U.S.A. (2000)
17. Duda, R. O., Hart, P. E., Stork, D. G. : Pattern Classification. John Wiley & Sons, New York (2001)
18. CONASIDA (ed.): El Médico Frente al SIDA. Pangea Editores, México (1989)

The use of Weighted Metric SOM Algorithm as a Visualization Tool for Demographic Studies

Elio Villaseñor, Humberto Carrillo, Nieves Martínez de la Escalera, and Valeria Millán

UNAM, Mexico City, Mexico

Abstract. Unsupervised neural networks provide a useful resource for exploratory data analysis. Here, we present an application of the SOM with weighted metric as an automatic tool to explore a large base of digital data of information about the student population of the National Autonomous University of Mexico,in order to look for gender differences impress in the academic performance. Our study proves the usefulness of this technique to visually analyze the performance and academic paths of several courts of students.

1 Introduction

Finding interesting structures and novel relations hidden in vast multidimensional data sets, be they textual documents, experimental data, or statistic information, is difficult and time-consuming. For these data mining tasks, information visualization techniques are valuable to display and analyze the discovered knowledge and therefore has become a topic of significant development and research.

The use of neural networks have proved to be useful in the data mining and knowledge discovery processes [1]. They are particularly valuable for the automatic generation of knowledge maps, that compactly convey information and are easy to analyze. By means of a "self-organizing" procedure, unsupervised neural networks based on mathematical algorithms, are capable of automatically explore large data bases. These neural networks are effective, not only to discover the structure of the data set, but also to reveal these patterns in a well organized knowledge map. This process involves two fundamental tasks: (i) the classification and cluster analysis; (ii) the geometric projection from the multidimensional space of data towards a two dimensional map.

The Self-Organizing Maps (SOM) algorithms [2] constitute a family of neural networks models that are widely used for exploratory data analysis and classification and clustering tasks. These algorithms due their popularity to the following facts: (i) by the use of reference vectors they allow an easy coding of multidimensional data that can be projected into a plane map by means of

Advances in Artificial Intelligence: Algorithms and Applications
Research in Computing Science 40, 2008, pp. 13-25

a topology preserving transformation of the multidimensional space; (ii) supervised training is not required and (iii) the high performance of the algorithms allow the treatment of large databases.

During the unsupervised training of the SOM algorithm the neural network recognizes the structure of the data set. The following face of the process involves a visualization technique to generate a knowledge map over the two dimensional array of neurons.

The use of a weighted metric play an important role to obtain a visually adequate representation of the information and knowledge contained in the map. In this paper we discuss and illustrate, in the context of demographic study, the usefulness of a *variable scaling technique.* In this application the use of a weighted metric introduces a competition criteria that incorporates a hierarchical order of the variables that constitute the multidimensional data space.

In this work we present an investigation using a SOM algorithm with the purpose of finding gender differences in the academic performance of students in the Universidad Nacional Autonoma de México (UNAM).

2 A SOM Based Visualization Technique

A Basic SOM is a training neural network model that consider a two dimensional regular processing grid of neurons $\mathcal{N}$, with a set of reference vectors (synaptic weights) $W = \{w_\eta\}_{\eta\in\mathcal{N}}$ such that $W \subseteq \mathbb{X}$. Where $(\mathbb{X}, d)$ is a n-dimensional metric space. For each $\eta \in \mathcal{N}$, its reference vector has the form $\omega_\eta = (\zeta_1^\eta, \zeta_2^\eta, ..., \zeta_n^\eta,)$. If we consider a data set $X \subseteq \mathbb{X}$, at the end of the SOM training process, it is defined a projection mapping of the form:

$$\varphi : X \to \mathcal{N}, \tag{1}$$

given by the condition

$$d(x, \omega_{\varphi(x)}) = \min_{\eta\in\mathcal{N}}\{d(x, \omega_\eta)\}.$$

The main property of 1 is named "topology preserving", i.e. if $x, y \in X$ are close in $\mathbb{X}$ then $\varphi(x)$ and $\varphi(y)$ are close in $\mathcal{N}$. *Topology preserving maps, were originally created as a visualization tool; enabling the representation of high-dimensional data sets onto two-dimensional maps and facilitating the human expert the interpretation of data* [3].

A SOM-based data visualization method consist in a coloring of the cells in the two dimensional SOM's grid $\mathcal{N}$. One example is the U-matrix. For each $\eta \in \mathcal{N}$ consider U_η a neighborhood of η and the average distance u_η between the reference vector ω_η and the reference vectors of the neurons in U_η:

$$u_\eta = \frac{1}{\#U_\eta} \sum_{\nu\in U_\eta} |\omega_\eta - \omega_\nu| . \tag{2}$$

Where $\#U_\eta$ is the cardinality of the neighborhood. In the U-matrix method, u_η is used as a measure of accumulation of similar data. Assuming the topology preserving property, the information given by the projections of the $\{u_\eta\}_{\eta\in\mathcal{N}}$ over

the grid $\mathcal{N}$, provides insight about similarity relations present in data.. For the visualization of this projection is used a bijection ϑ among $[\min\{u_\eta\}_{\eta\in\mathcal{N}}, \max\{u_\eta\}_{\eta\in\mathcal{N}}]$ and a monochromatic (e.g. gray-scale) color bar. For each $\eta \in \mathcal{N}$, its color is given by $\vartheta(u_\eta)$.

Although, the information that gives the U-matrix concerns only to the clusters structure present in the data but does not give information about the correlations between data variables. By studying this correlations it is possible to find out causality relations among data components. One technique that is useful to find correlations between variables is called Component Planes.

For each component, consider its values $\{\zeta_k^\eta\}_{\eta\in\mathcal{N}}$ on the the grid, the ζ_k – *Plane* is a coloring given for a bijection ϑ_k between the interval $[\min_{\eta\in\mathcal{N}}\{\zeta_k^\eta\}, \max_{\eta\in\mathcal{N}}\{\zeta_k^\eta\}]$ and a chromatic color bar. Therefore each variable ζ_k of the data set is projected using a coloring ζ_k – *Plane*. The comparison of two distinct component planes is useful for finding correlations of corresponding variables. *Correlations between component pairs are reveled as similar patterns in identical positions of the component planes. Pattern matching is something that the human eye is very good at* ...[4]. In the section of computational results these kind of analysis are presented.

3 Weighted Metric in the Multidimensional Space

The quantitative and qualitative information contained in the input data set has to be mathematically modelled in order to automatically explore the database. For this, each individual data is represented by a vector in an abstract multi-dimensional space. To analyze the structure of the *row data set* the use of a metric is in order. Typically the standard Euclidean (homogeneous) metric is used, however in certain applications it is convenient to consider an alternative metric in the row data set. The use of weighted metrics is a convenient resource to incorporate an element of competition during the training process, among the variables that represent the data in the space $\mathbb{R}^n$. Accordingly, the associated weights, pondering the relative relevance of the individual variables, establish a hierarchy in the multidimensional space.

A weighted metric is a function $d : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$. For each $x, y \in \mathbb{R}^n$ with components $x = (x_1, ..., x_n)$ and $y = (y_1, ..., y_n)$ the weighted distance between x and y is given by

$$d(x, y) = \sqrt[2]{\sum_{k=1}^{n}(w_i(x_i - y_i))^2},$$

where $w = (w_1, ..., w_n)$ is the vector of weights associated to each dimension.

The use of a weighted metric affects the induced projection, φ, from the row data set, $X \subset \mathbb{R}^n$, to the two dimensional neural grid, $\mathcal{N}$. The way in which the global ordering of the points in X projects over the $\mathcal{N}$ grid is determined principally by those components with heavier weights meanwhile the more local relations of similarity of the data are ordered in $\mathcal{N}$ by those components with

lighter weights. Thus the training might differentiate first regions that correspond to the variables with the heavier weights and a further differentiation of these regions will subdivide them in subregions determined during the training by the variables with lighter weights.

In the following section, we illustrate the application of the SOM algorithm with a weighted metric in the frame of a demographic study of five student cohorts (39,893 students) of the UNAM. In this study each student is represented by a 28-dimensional vector, according to the order of magnitude of the associated weights to each of the vector components, are divided in three classes. One of the vector components is the variable sex to which, due to the purposes of our investigation, we have assigned the largest weight (2). The effect of this weighted metric in the projection map is the identification of two clearly separated gender regions (figure 2(a)). We point out how, inside these two regions, the neural net finds difference patterns, for each gender, associated to the 20 variables that measure the career-progress percentage of the students; a weight of 0.1 was applied to these variables. Using these twenty components we calculate a new discrete variable of the data vector that constitutes an indicator of each student's final status on finishing her or his studies: *Normative Graduation, Deadline Graduation, Terminal Graduation and Dropout*; a weight of 0 was applied to this variable. The next component correspond to the area of knowledge in which the career selected by the students is classified: *Physics, Mathematics and Engineering, Biological and Health Sciences, Social Sciences and Arts and Humanities*; this variable was weighted by 0.01. Finally the five following components contain the information on the quantified categorical variables, which, in our investigation, constitute the presumed achievement factors: marital status, children, job, the mother's academic background, whether the student's family owns a car or not, etc. We give this variables a weight of 0.01.

4 Practical Application: A gender study of student population of UNAM

As shown in figure 1, during the1980-2005 period the UNAM student population became stable with approximately 140,000 students. However, during the last 25 years there has been an important change in the sex ratio. In 1980, the male population was twice as large as the female population. From then on, the female population has increased notoriously, while male population has decreased at the same rate. Because of this sustained trend, in 1994 the two populations became equal with values around 70,000 until 1999. During this time a strike that lasted almost a year paralyzed the Institution and made enrollment of a whole generation impossible, which resulted in a momentary reduction of the population. After this numerical fall, both populations have recovered, reaching figures similar to those before the strike, where an interesting phenomenon can be observed: the recovering rate of female population notably overcomes that of the male. From the year 2000, this sex ratio has kept up the disparity, and

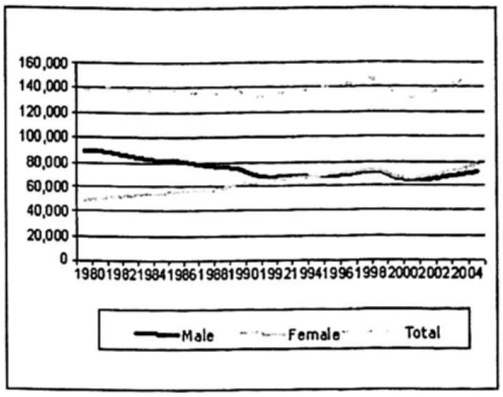


Figure 1. Dinamics of students population at UNAM.

in 2005, male population reached around 73,000, which was overcome by the female population of 80,000. If we calculate the change of these two population during the 1980-2005 period, there is a decrease in male population of over 25% and a growth of female population in more than 40%.

This behavior, [5] show: "Different gender social values and cultural norms based on masculine and feminine identity and specificity, as well as inequity conditions between the sexes are matters built and socially reproduced in different ways and in a dynamic manner". This phenomenon of substituting the male space by the feminine one should be studied because it is not the result of any special institutional effort to support this sector.

In a recent comparative study [6], about college education and gender in Latin America and the Caribbean, 'feminine' and 'masculine' careers are identified and typically associated to social roles and gender. These roles affect individual decisions and may create cultural barriers preventing an equal enrollment in higher levels as well as in the job market. As stated in a [7] study on Education in the XXI Century: "teaching and education are a source of social segregation according to gender, to the extent that the selection of professional and career paths are usually made before entering the job market".

In some articles, the UNAM case has been given particular importance. In the study [8] coordinated by the Programa Universitario de Estudios de Género (PUEG) entitled "The Presence of Men and Women in the UNAM: an X ray" gives a panoramic view of differences between women and men in three sectors of the UNAM: students, academicians, and administrative staff. Among the first results for the student population, the existence of careers that may be classified as typically masculine or feminine was validated. The evolution of gender predominance and detected variations tending to feminization is also analyzed. Besides, a more favorable positioning of women regarding the speed of curricular progress, higher averages and higher subject approval is noted.

In another gender study, social background and performance in the UNAM, [9] analyzes an enrollment cohort in 1997 and shows a consistent correlation between school performance and social factors. It can be concluded from the study that those differences noted regarding performance in individuals cannot be understood only as a product of innate skills but also as a product of other advantages and disadvantages which have an accumulative effect, among which the gender factor plays a complex role. In this study the importance of 'double shift' is also analyzed, the combination of study and work (generally males with salaries and women doing unpaid housework), and even in this scenario a better female academic performance and profit has being shown.

The database compiled for the analysis was obtained from two complementary sources: the academic record and questionnaires answered by candidates to the institution (Encuesta de Aspirantes de ingreso a licenciatura por concurso de selección y pase reglamentado). We discarded all those students who did not answer the questionnaire or did not answer one of the questions regarding our study. The consolidated sample has a total of 39,893 students which represents 59.2% of the population from the five studied cohorts. We analyzed the progress of these five cohorts for 20 semesters (most of the careers are planned for 10 semesters). After 20 semesters very few students finish their careers. In this experiment we get the same conclusions of the mentioned works by visually analyze the component maps of a SOM training with this database.

4.1 Computational Results

The following maps are conformed by a flat square hexagonal grid of 3,969 neurons, following the recommendation of Kohonen for the number of neurons to use one order of magnitude less than the cardinality of X [2]. We used a SOM algorithm implemented in ViBlioSOM system that is the Batch-Map variation with a Gaussian neighborhood function with a lineal radius decreasing. This is a software system that is in development by the "Laboratorio de Dinámica No-Lineal" of the Sciences School at UNAM. This system implements a SOM algorithm for the visualization of bibliometric information. In order to execute exploratory data analysis, taxonomy studies or Clustering, as well as generate cartographies through projecting data from the multidimensional space in a plane. This system is useful for informetric analysis who produce automatic knowledge maps representing the structure and information included in the data basis. The system also produces quantitative results and offers a variety of graphic scenarios for their representation. In this work, we only use the SOM-Based visualization capabilities of the system. In fact. the results of this applications are already presented in [10].

For example, Sex-Plane (figure 2(a)) we established a chromatic spectrum that goes from blue to red (as in rainbows). In this case, the red color represents women, and blue represents men. Thus, in this map we can identify two distinct areas (clusters of cells) in which men and women have been distributed. Both, the red and blue regions of this map have been clearly differentiated by a diagonal division. We call these regions the feminine zone and the masculine

zone. They both span areas that are almost equal in size within the neuronal grid which has been created by the map, although the feminine zone is slightly larger than the masculine zone. This suggests that the feminine population is slightly larger than the masculine one, and this reflects a numeric reality: there are 22,127 women, whereas there are only 17,765 men.

In the Performance-plane (figure 2(b)), this process has created four different zones. These four classes are determined according to the time in which the students covered an equal or superior percentage of 90% of the semesters which are required in order to finish their Bachelor's Degree studies. The term Normative Graduation corresponds to both male and female students who accredited at least 90% of their studies during the first 10 semesters. The term Deadline Graduation corresponds to the period of time in which this 90% percentage of credits has been reached between the 10th and 15th semesters. The term Terminal Graduation corresponds to students who have finished their studies between the 15th and 20th semesters. And the term Dropout corresponds to students who do not manage to finish their studies, and thus do not graduate, even after being registered at UNAM for 20 semesters. We give a weight of 0 to this component because it has been considered in the previous twenty components. The Normative Graduation zone (which is red) corresponds to both male and female students who have "finished or completed" their career studies (having covered 90% of their credits) during the allotted time. In similar fashion, the Deadline Graduation zone has been highlighted in green, the Terminal Graduation zone has been highlighted in yellow, and the Dropout zone has been highlighted in blue. Any one can clearly see that this map is quite symmetrical in relation to the diagonal line which descends from left to right. Nevertheless, the Normative Graduation zone (highlighted in red) covers a greater extension of space in the feminine region. Besides, one can observe that the Deadline Graduation zone (yellow), the Terminal Graduation zone (green), and the Dropout zone (blue) cover areas of similar size in the masculine area as well as in the feminine one.

In Job-plane (figure 3(a)), we can perceive an outstanding asymmetry: the yellow and red tonalities predominate (these correspond to students who were working at the time when they answered the questionnaire) in the masculine zone. The greater part of these cases appear distributed in the Dropout zone. The color red does not appear in the feminine zone, but we can observe green regions. As the color of the cells reflects the average value of the component which is being analyzed, one must interpret that in the green cells (which stand between the red and the blue zones) we have grouped both the women who hold jobs as well as those who do not. It must be noted that these tonalities predominate in the Dropout zone.

In the Children-plane (figure 3(b)), blue regions indicate the place within the map held by students who already had children when they begun their studies. The greenish-yellowish tonalities are distributed homogeneously across the map, but we can observe that these colors predominate in the feminine Dropout zone, with some exceptional cases in the feminine Normative Graduation zone. The Marriage Status-plane (figure 3(c)) shows a very similar distribution pattern that is quite similar to the previous component plane.

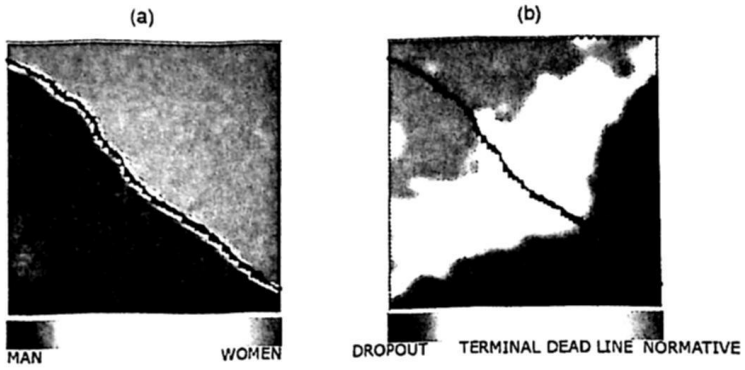


Figure 2. (a). Sex-plane (b). Performance-plane

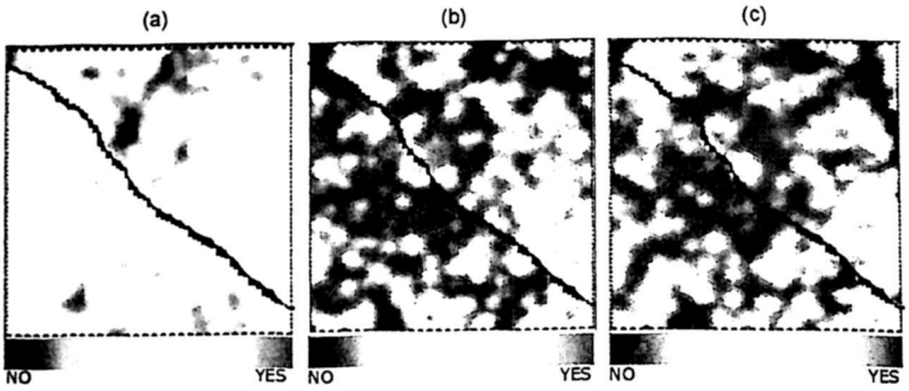


Figure 3. (a). Job-plane (b). Marriage Status-plane (c). Childen-plane

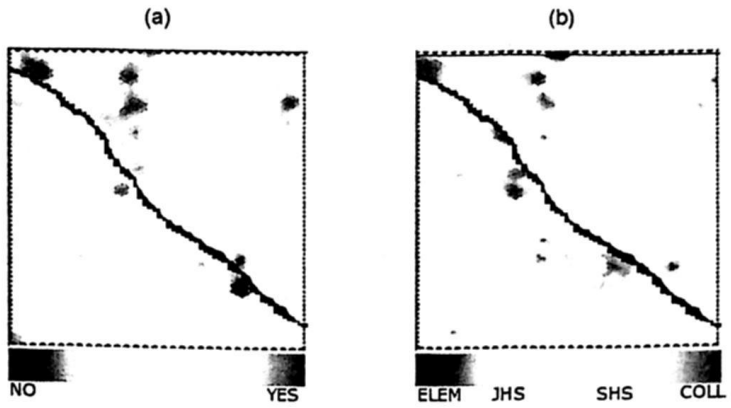


Figure 4. (a). Automovil-plane. (b). Mother. Instruction-plane

In the stratification of Mother Instruction-plane (figure 4(b)): The Mother's Academic Background, we can see a greater concentration of yellowish-reddish blots (students whose mothers have a higher academic level) can be seen in the feminine Normative Graduation zone. This same type of organization is manifested in Automovil-plane (figure 4(a)): Car, although they are more dispersed throughout the whole map, and there are some yellowish-reddish blots in the masculine Normative Graduation zone.

The distribution of colors that correspond to the Areas of Knowledge in which students chose their career studies (Area-plane figure 5) is less disperse if compared to previous maps. However for the correct interpretation for this component plane, it has to be considered that there are neurons that do not have the exact color that is assigned to an specific area. In this neurons, it is sure that there are students whose career do not belong to the same area of knowledge. Taken into account this considerations, it is noteworthy that the region that corresponds to Physical and Mathematical Sciences as well as Engineering (blue) is made up of two unrelated conglomerates. This highlights the existence of two different classes of typical trajectories within this subpopulation group, and which should be conceptualized. One of them inhabits the Normative Graduation zone with a notoriously dominant component in the masculine zone. The second blue conglomerate is slenderer than the previous one, and is enclosed, almost completely, in the masculine Dropout zone. Much in the same manner, in the feminine Normative Graduation zone we can see a red conglomerate which corresponds to female students belonging to Arts, Philosophy and Literature academic careers. In this same zone we can also observe the dominant presence of yellow, which corresponds to the Social Science area.

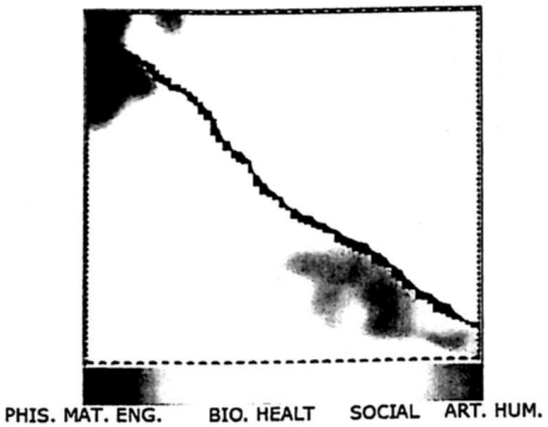


Figure 5. Area-plane

4.2 Discussion and Interpretation

The computational results we have obtained by means of the ViBlioSOM system are consistent with available information as well as with the results of previous investigations. This validates the procedure we have followed.

This experimental investigation has served us well in verifying the different work hypothesis, thus confirming the prevalence of noteworthy gender differences in academic achievement. These differences are clearly visible within the different time modes in which students manage to finish their studies, or not, as well as in the different career studies that students in the UNAM choose to study:

a. If the categorical variables we have proposed were not legitimate achievement factors, the regions of different colors which appear on the component maps would not fit regularly, as we have already seen, within the different Graduation and Dropout zones.

b. On the other hand, given the almost perfect symmetry that can be seen in the zones pertaining to gender classification, if any of the proposed achievement factors did not have a differentiated effect on gender issues or if students' preferences when choosing the area of knowledge in which they wish to study would not be marked by the difference in gender, one could expect that the coloring in the corresponding component maps would be distributed in a symmetrical manner in reference to the diagonal line that divides the feminine zone from the masculine one.

Gender differences were expressed in the creation of the maps in the following

manner:

- The map of the Gender component clearly shows the feminization of the enrollment pattern within the UNAM. This could mean a progress towards gender equality and it also constitutes a potential factor that can contribute to the balance of opportunities for both female and male graduates when looking for a job and as an equal means to achieve social standing.

- From the analysis of Areas of Knowledge map, we can confirm the presence of academic spaces which are considered typically masculine, such as Physics, Mathematics, Engineering, as well as others which are considered typically feminine, such as Arts, Philosophy and Literature. One possible explanation for the dominance of women in the Normative Graduation zone is the "double shift" characteristic, which in the case of men implies holding a job besides studying. This conjecture is supported by the information obtained from the Job map: when students begin working at an early age, this can have a negative effect on the time they take to graduate from the university, or it even forces them to finally drop out. Thus we can conclude that for men holding a job at an early age, compelled to earn money in order to assume their traditional gender role, their role as breadwinners competes with their role as students, making them lag behind in their studies.

- Entering the university either alredy married or with children is not a common condition within the freshman population (freshman average age is 19.97 years, std. deviation 3:02; men 20:30 years, std. deviation 3.09; women 19.72 years, std. deviation 2.93). As is to be expected, both the Marital Status and the Children maps coincide quite a bit. These maps reveal a differentiated tendency by gender for those students whom, upon enrolling were already married or already had children: in the masculine zone, the scarce presence of students married or with children does not have a notorious influence on graduation timing, and it is distributed evenly in the different graduation zones. On the other hand, married women or women who have children are concentrated preferentially in the Dropout zones. This is reminiscent of the traditional roles played by both genders, according to which women must take care of their homes and their families, while all professional aspirations are deposited in men.

- An additional confirmation of the validation of the methodology we have used is offered by The Mother's Academic Background Map and the Car map. Both express the supposed relationship between the socioeconomic status of students and their academic achievements. It is commonly thought that a higher socioeconomic standing will necessarily mean a competitive advantage for academic performance. This is confirmed in maps which show a prevalence of yellow and red spots in the Normative Graduation zone.

- Furthermore, the exploratory analysis of data we have carried out by means of the ViBlioSOM system allows us to make the following predictions, which should be the object of future investigations:
  - Economic status carries less weight in graduating during the normative time than the mother's academic level.
  - The fact that a student's mother has a high academic level can certainly have an outstandingly positive effect on the academic trajectory of female students, but not so much in the case of male students.
  - The differentiated effect that a mother's academic achievements have on female students, together with the current process of substituting masculine spaces by female ones, clearly suggests that this feminization trend will be maintained in the long term. This conjecture must become the subject for future research.

5 Conclusions

In this work we present an experimental demographic application of the basic SOM model with weighted metric. The interpretation of the maps coincide with the results of other researches of gender differences at the UNAM and new hypothesis were formulated. With the puropose of getting meaningful maps it is useful to incorporate hierarchical orders in the variables via weighting of variables. Once the logic principle of map's interpretation is understood, the visual analysis is very easy to be done. So it is possible to get very valuable information by simply viewing the maps without considering statistics. We think that this kind of applications can be extended to other demographic studies and it can be a great support to many important decisions in social researches.

Acknowledgement *"Macroproyecto Tecnologías para la Universidad de la Información y la Computación".*

References

1. Bigus J., "Data Mining with neural networks", McGraw Hill, USA, 1996. Buquet, Anaetal. Presencia de Mujeres y Hombres en la UNAM: una radiografía. Programa Universitario de Estudios de Género (PUEG)-UNAM, México (2006).
2. Coñeen T., "Self-Organizing Maps", 3ra Edición, Springer-Verlag, 2001.
3. Baruque B., Corchado E., Yin H., "ViSOM Ensambles for Visualization and Classification", F. Sadoval et al. (Eds.): IWANN 2007, LNCS 4507, Springer-Verlag 2007.
4. J. Vesanto, "SOM-based data visualization methods," Intelligent Data Analysis, vol. 3, April 1999.
5. Szasz, I. and Susana L. "Aportes teóricos y desafíos metodológicos de la perspectiva de género para el análisis de los fenómenos demográficos" in

Susana Lernery Alejandro I. Canales (eds.). Desafíos teórico-metodológicos en los estudios de población en el inicio del milenio. COLMEX, UDG, SOMEDE, México (2003), pp.177-209.

6. Papadópulos, Jorge and Radakovich, Rosario. "Estudio comparado de edu-cación superior y género en América Latina y el Caribe" Educación superior y género en América Latina y el Caribe, IESALC, Unión de Universidades de América Latina-UDUAL.(2003). Ch. 9, pp.118-128.
7. Internacional Labour Organization. La educación permanente en el siglo XXI: nuevas formas para el personal de educación, Geneve (1998).
8. Buquet, Anaetal. Presencia de Mujeres y Hombres en la UNAM: una ra-diografía. Programa Universitario de Estudios de Género (PUEG)-UNAM, México(2006).
9. Mingo, Araceli. ¿Quién mordió la manzana? Sexo, origen social y desem-peño en la universidad, Universidad Nacional Autónoma de México-Fondo de Cultura Económica (2006).
10. Millán V., Villaseñor E., Martínez de la Escalera N and Carrillo H, "Informetrical Visualization of Gender Differences in College Performance: an application of ViBlioSOM", sended to Resources for Feminist Research.

Determinants of Export Performance: An Analysis using the SOM Algorithm

Omar Neme[1], Antonio Neme[2], Alejandra Cervera[3]

[1] Sección de Estudios de Posgrado-Escuela Superior de Economía, Instituto Politécnico Nacional
[2] Non-linear dynamics and complex systems group, Universidad Autónoma de la Ciudad de México
[3] Comisión Nacional para el Uso y Conocimiento de la Biodiversidad, México.

Abstract. Mexican international trade growth has been characterized by the relative success of some industries in the US market. In order to analyze determinants of the Mexican export performance, we use a different methodology: the self-organizing map (SOM), which approximates the distribution observed by the economic variables in the original high dimensional space. Consequently, we used the SOM for studying the Mexican exports from 1985 to 2006, at industry level, classified according to their technological intensities. Each year is represented as a vector of 80 components. SOM reflects the similarities and differences in the way exports could be affected by this variables. The results show the efficacy of this approach in explaining the export performance of manufacturing industries. Particularly, the SOM reflects that industries are distributed along the map confirming the existence of notable disparities among them. Moreover, some industries appear relatively near which allow us to suggest these industries, and some of its firms, share important characteristics in terms of export success.

1 Introduction

The self-organizing map (SOM) is a non-linear projection from a multidimensional space to a discrete low-dimensional space. The projection achieved by the SOM is an approximation of the distribution observed by the points in the original high dimensional space [1]. The low dimensional space is a lattice of neurons. Each multidimensional data is mapped to one neuron known as the Best Matching Unit (BMU) for that input data. The set of BMUs from all neurons shows the distribution achieved by the SOM of the multidimensional inputs. Objects located close to each other in the feature space, will be mapped to nearby neurons whereas distant objects will be mapped to distant neurons.

The self-organizing map (SOM) is a model of self-organization of neural connections, which reflects the ability of the algorithm to produce organization from disorder [2]. One of the main properties of the SOM is the ability to preserve in the output map those topographical relations present in the input data [1], a very desirable property for data visualization and clustering. This property is achieved

Advances in Artificial Intelligence: Algorithms and Applications
Research in Computing Science 40, 2008, pp. 27-37

through a transformation of an incoming signal pattern of arbitrary dimension into a low-dimensional discrete map (usually one or two-dimensional) and by adaptively transforming data in a topologically ordered fashion [1, 3]. Each input data is mapped to a single neuron in the lattice,to the one with the closest weight vector to the input vector, or BMU. The SOM preserves neighborhood relationships during training through the learning equation, which establishes the effect each BMU has in any other neuron. The weights of the neurons are updated accordingly to:

$$w_n(t+1) = w_n(t) + \alpha_n(t)h_n(g,t)(x_i - w_n(t)) \quad (1)$$

where $\alpha(t)$ is the learning rate at time t, $h_n(g,t)$ is the neighborhood function from BMU, neuron g, to neuron n at time t, and x_i is the inpyt vector. In general, the neighborhood decreases monotonically as a function of the distance between neuron g and neuron n. The SOM tries to preserve relationships in the input data by starting with a large neighborhood and reducing it during the course of training [1].

The paper is organized as follows. The next section explains with more detail the SOM algorithm used to analyze the export performance. Section three presents the results of applying the SOM methodology to some micro and macro variables related to the export performance. Finally, we present the conclusions of this study.

2 Related work and Proposal

SOM has been widely applied as a visualizing tool in different areas. For an introduction and in-perspective analysis of the SOM as a visualizing tool, see [4, 5]. In economics, the SOM has also been a valuable tool for data analysis and forecasting. In the seminal work of Kaski and Kohonen [6], the social welfare of 39 countries is analyzed at the light of 9 variables. Since then, several related works have emerged, and interesting results have been reported [7].

Here, we apply the SOM for studying the Mexican exportations from 1985 to 2006. Each year is featured as a vector of 80 components, where each component corresponds to a given type of exportation (goods or services), Through SOM, we observe similarities and differences in the way this important economic component has been affected.

Mexican international trade growth has been characterized by the export of manufactures of diverse quality and characteristics, and by elevated imports of inputs to be processed and then re-exported. In this way, Mexico has become an important supplier of exports for international markets, in particular for the United States (US).

In this context, the strategy for economic growth has been based, mainly, in the promotion of relatively high intensity technology manufacturing exports. As a result, the weight of exports with respect to the gross domestic product is today near 45%, showing an annual average growth rate of 50%. At the same time,

public and private consume spendings and the fixed capital gross investment lost their relevance as sources of economic growth. In the same way, economic growth registered an annual rate of 5% in this period. In consequence, it is accepted that the notable export expansion, given the small rate of change of internal market, is the responsible of that economic growth.

However, manufacturing export industries development has been unbalanced in comparison with the competitive advantages that each one enjoys. The Mexican export specialization is, according to the Organization for Economic Cooperation and Development (OECD) [see www.oecd.org] data, in medium-high and high technology industries, among these are motor vehicles, communication equipment and computing machinery. In contrast, the sectors that reveal the smallest advantages correspond to labor intensive such as textiles, apparel and footwear, which employ capital in a low proportion. But, which factor determines the level of industry exports? Do technology, market structure, foreign demand, costs and other variables affect the success of industries exportation? Do industries share similar characteristic that allow them to export? The answers of these questions have important considerations of industrial policy.

Classification of industries by common characteristics allows a better understanding of the industries behavior and consequently better planning of sectoral politics with the objective of improving, in a coordinated manner, the industrial structure.

On the other hand, foreign trade, owing to new foreign trade theories, can be explained through a multidimensional relationship. Here, the central variables, market structure and technological innovation process, allow domestic industries to reduce average costs as the production level grows, to manufacture new designs and to promote projects that involve research and development with the goal of introducing new differentiated products in international markets. Other aspects like human capital, regional integration and spillover effects derived from foreign technology, play an important role in export performance too.

In this manner, we argue that potential determinants of export performance (or alternatively export competitiveness) of the manufacturing sector in the Mexico-US relationship are gathered in two. In one group, we have the traditional price competitiveness, that includes prices such as real exchange rate or relative unitary cost labor and, on the other, the non-price competitiveness, including innovation processes, domestic production capacity, external demand, scale economies and product differentiation. Likewise, we accept that both kinds of competitiveness affect in different ways each industry.

In this context, based in an abundant set of variables associated with different aspects of the export process, the objective of this paper is twofold. The first is to apply the SOM methodology to the economic analysis and, the second, is to represent the relationships among several potential determinants of foreign trade for each industry, allowing us to group or classify exporting industries considering multiple dimensions. So, by using the SOM algorithm, high-dimensional data can be projected to a lower dimension representation scheme to facilitate the economic analysis.

In this regard, econometric methodology offers several options for estimating the export function, for determining relationships among variables, for simulating politics and for forecasting new relationships. Among econometrics alternatives exist the one equation regression model, the multi-equational models, and time series analysis that have evolved from classic methods to modern techniques that let the researcher determine long run equilibrium relationships. The utilization of this methods in the pursue of our goals implies numerous suppositions, some of them not belonging to the economic theory. For example, it is required to establish some nullity restrictions over the parameters, the type of relationship among variables, sign and nature, and the estimation method, amid others. Simultaneously, these procedures, based on the assumption of linearity, provide us with quantitative arguments regarding the relationships among variables, but they are considerably restrictive and do not necessarily reflect the reality. Hence, modeling with non-linear algorithms seems more appropiate to the actual study.

It is possible to model the export performance through alternative methods like the Self-Organizing Map (SOM). The SOM algorithm, just like econometrics, allows the estimation of relevant relationships between Mexican industries export performance and the variables gathered in price and non-price competitiveness.

Moreover, using the SOM gives additional advantages like the graphic representation in two dimensions of those multivariate relations, which, in turn, allows us to group the international industries in accordance with the common characteristics, from an alternative perspective to the factorial analysis or main components. In other words, through the SOM it can be represented in a map of export performance the state or level of the export success of each industry, recognizing the grade of similarity where potential determinants influence the commercial flows in the industries.

The high number of variables involved in the determination of manufacturing exports make it difficult to analyze and to draw conclusions of interest for the policy makers. With SOM, we expect to greatly simplify the identification of central variables affecting export flows of Mexican industries.

3 Results

The SOM ordered the Mexican exporting industries, as it can be seen in Figures 1a and 1b for the 1985 and 2006 years, respectively. These figures can be interpreted as a map of sectoral export performance although no information of this kind was included in computing and training the SOM. The SOMs show the way industries behave individually as a consequence of different micro and macro economic variables. Several aspects can be inferred from their distribution in the maps. First, in 1985, there is no clear technological pattern of the considered industries; since some low-tech industries, Wood Products and Furniture (20), or medium low-tech, Other Non-metallic Mineral Products (26), are very close to some high-tech industries, Computing Machinery (30) and Pharmaceuticals (2423), in terms of export performance (see table 1 for the complete list of industries).

Table 1. Identification for all industries.

Label	Industry
15-37	Total
15-16	Food products, beverages, and tobacco
17-19	Textiles, textile products, leather, and footwear
20	Wood and products of wood and cork
21-22	Paper, Paper Products, and Printing
23	Coke, refined petroleum products and nuclear fuel
24-2423	Chemicals excluding Pharmaceuticals
2423	Pharmaceuticals
25	Rubber and plastics products
26	Other non-metallic mineral products
27	Basic metals
28	Fabricated metal products, except machinery and equipment
29	Machinery and equipment, n.e.c.
30	Office, accounting, and computing machinery
31	Electrical machinery and apparatus, n.e.c.
32	Radio, television, and communication equipment
33	Medical, precision and optical instruments, watches and clocks
34	Motor vehicles, trailers and semi-trailers
351	Building and repairing of ships and boats
353	Aircraft and spacecraft
352-359	Railroad equipment and transport equipment n.e.c.
36-37	Other Manufacturing

This important result implies that the technological factor is not a main determinant of the export competitiveness of the Mexican firms in that year. That is, despite the differences in innovation capacities among industries (like human capital, import of technology, licenses, or patents) that shape technological advantages in the foreign markets, other variables such as scale economies, market structure, product differentiation, production capacity, foreign investment, labor cost, or external demand influence the performance of these industries in the US market. Moreover, although industries are classified in function of their technological intensities, the Mexican manufacturing industry employed intensively the labor factor even in sectors considered as technologically advanced. Then, we can affirm that differences in the function production, by definition of industry, are counterbalanced by the combination of labor and capital factors in their system of production. Finally, this argument implies a delayed technological competitiveness of the exporting industries. Besides, industries seem to distribute, in general, along the map, implying notable disparities. Distances among industries are relatively large. In particular, low and medium-low technology industries are, in all the cases, considerably separated,by at least two cells; situation we explain with the following arguments. First, the internal capacities of the firms allow them to face the technological opportunities in different ways and with different results. Second, external relations derived from the market led them by diverse

growth paths. It can be observed that the SOM algorithm grouped high and medium-high tech industries relatively closer among themselves, as it could be expected due to the technological variables considered in this analysis, which in a way confirms the OECD taxonomy [see www.oecd.org]. Inside the first category, Pharmaceuticals (2423), Computing machinery (30) and Medical and precision and optical instruments (33) keep neighborhood relations; as well as Motor vehicles (34) and Chemicals excluding Pharmaceuticals (24-2423) inside the second group of industries. Following the argument presented above we can assume that technological gaps among these industries are small . This means that industries share similar characteristics in terms of innovation capacities, corporative strategies and are influenced approximately in the same way by the structure market and other variables. Also, the SOM shows how the manufacturing sectors are affected by a set of variables; now we are interested in the individual influence of each variable on each industry. Variations in most of the indicators have not a clearly distinguishable pattern; consequently the distribution of the industries in the SOM reflects this issue. Considering just some fundamental variables, we could assume, for example, that firms market power -oligopolic structure- boost the export performance as new international trade theory proposes(see [8]). Figure 2a reflects how the industries with major exports values are mapped almost together in a column located in the right side of the map. The exporting industries with relative poorer performance in the US market appear in the left side and in the right inferior co rner of the map. Another variable that influences export competitiveness is product differentiation. Figure 2b shows the distribution of the industries in accordance with this indicator. A relatively similar structure emerges from this dimension, that is, we can see a column in the right side of the map; although three industries with high values of product innovation are mapped far of them. As a result, we can state that product differentiation -originated from regular innovation activities- and imperfect market structures are two main determinants of the export performance in a context of high dimensional data inside the US market. On the other hand, external demand and foreign direct investment (FDI) influence the level of exports in the US market in a similar form. The industries are mapped in roughly the same way. Looking at Figures 2c and 2d we can distinguish a cluster of industries located in the bottom part. Also, Figure 2d has another cluster in the right flank, medium-seized. With the values of the exports of the industries included in these clusters, it could be argued that imports and FDI are a second kind of determinants of export performance. Figures 2e and 2f show the cost competitiveness relative unitary labor cost and the production capacity both in the manufacturing sector. The former suggests how cost dimension negatively affects all the industries in terms of their exports except for Other manufacturing (36+37). For example, the export level of Computing machinery (30) was restricted by its low cost advantage derived from the low capital intensity in relation with the intensity in the US market. The latter, although important for some industries, does not appear to be a central factor for the exports; some industries like Coke and refined petroleum products (23), Wood products and furniture (20), Building and

repairing of ships and boats (351) and Railroad transport equipment (352+359) move in the same direction of production capacity in the domestic economy. Hence, costs and product capacity are fundamental determinants of the manufacturing exports although the dimension of the map does not seem to reflect the exporting success of all the industries. Finally, we consider the spillover effects -a concept extremely linked with the FDI- over the exporting industries (Figure 2g). This indicator is a source of competitiveness for Fabricated metal products, except Machinery and equipment (28), which means that foreign capital invested in the Mexican economy into this sector has a double impact. First, FDI provides the production and export capacities and simultaneously it gives other kind of advantages such as learning by doing, learning by exporting, technology, etc., that impact in the productivity level of firms in this industry. Nevertheless, once again, the greater effects of these indicators do not occur in the major exporting industries. On the other hand, in 2006, in contrast with 1985, it seems that the industries are mostly grouped in accordance the OECD technological classification. That is, industries with the same technological intensity are relatively close among them (see Figure 1b). Hence, comparing the SOMs, it can be stated that with time a more visible technological pattern of exporting industries emerged. However, there is more dispersion of the sectors and some are mixed with other industries that exhibits different technological capacities. This result implies that in 2006, the technological factor is more important in the export competitiveness of the Mexican firms than it was in 1985. Therefore, although other variables mentioned above -market structure, product differentiation, labor cost, external demand, etc.- influence the success of exporting industries, there are differences in innovation capacities among them that have become the sources of that competitiveness. In other words, the distribution in the SOM suggests that Mexican manufacturing industries employ intensively the capital factor in the sectors considered as technologically advanced; while labor factor is the central input in the less technological developed industries. As a consequence, we can assume that technological competitiveness of the high and medium high technological sectors grew unlike medium low and low tech sectors. Besides, just like in 1985, industries, in general, seem to distribute along the map implying notable disparities; but, in contrast, distances among them are relatively less extensive. In particular, high and medium-high technology industries are significantly closer or, more exactly, are in the same zone of the map; which implies industries have reacted to the technological change and international conditions (like product relocalization, the merge of new industrialized countries, etc.) in the same way, allowing them to obtain a similar export success. As previously stated, the SOM algorithm grouped high and medium-high tech industries relatively closer among themselves. Inside the first group are Pharmaceuticals (2423), Computing machinery (30), Medical and precision and optical instruments (33) and Aircraft and spacecraft transport equipment (353); while Motor vehicles (34), Chemicals excluding Pharmaceuticals (24-2423) and Machinery and equipment (29) are inside the second one. Again, we can assume that industries share similar characteristics in terms of innovation capacities, corporative strategies and are

probably influenced in the same way by some indicator associated with market structure, cost or foreign demand. Apart, practically all of the indicators have different effects over the industries; although variations in most of the indicators have not a clearly distinguishable pattern. Product differentiation determines export competitiveness (Figure 3a), where the distribution of the industries allows us to distinguish a big group of them affected similarly by innovation product. Nevertheless, medium-high and high technology industries seem to be influenced more by this indicator than any other industry. Again, Figure 3a, reflects how the industries with major exports values are mapped more or less together at the bottom and at the left part of the map, leaving three isolated industries (30, 351, and 352+359). The exporting industries with relative poorer performance in the US market appear in the rest of the map. In opposition, market power appears to influence positively just a few industries -Building and repairing of ships and boats (351), Railroad transport equipment (352+359), Spacecraft transport equipment (353), Computing machinery (30), Electrical machinery apparatus (34), and Radio, television, and communication equipment (32), have an advantage in the US market. Accordingly with the SOM (Figure 3b), the industries where strong oligopolic competence exists are located at both, right and left, upper corners and at the central part of the map. This kind of market structure can lead the major exporters to price discrimination practices, that is, to establish higher prices in the domestic market and lower prices in the foreign ones, as a strategy to access the international market. Nonetheless in a big number of industries, perfect competition operates. Thus, we could assume that firms market power rise the export performance as well as product differentiation. On the other hand, the SOM suggests that the level of exports is directly linked to external demand and foreign direct investment in relatively few industries. Figure 3c distinguishes a cluster of industries located at the lower left corner. These say that US demand conditions influences mainly the exports of Motor vehicles (34), Radio, television, and communication equipment (32), Machinery and equipment (29) and Textiles, apparel, leather, and footwear (17+19). In contrast, FDI affects positively Motor vehicles (34), Machinery and equipment (29), Chemicals excluding Pharmaceuticals (24-2423) and Food, beverages, and tobacco (15+16), as it can be seen in Figure 3d, where one cluster located in the bottom of the SOM, towards the left side, stands out. Alternatively, as we can expect, both indicators have a limited impact in the export performance boosting not just few industries, but also the value of their exports. Figures 3e and 3f show the relative unitary labor cost and the capacity production. The former, suggests cost dimension negatively affects all the industries in terms of their exports (except three industries 17+19, 30, 32). In particular, it shows one cluster formed by Coke and refined petroleum products (23), Rubber products (25), Fabricated metal products (28) and Chemicals excluding Pharmaceuticals (24-2423). In contrast with the 1985 case, apparently fewer industries are affected by this variable, suggesting that cost competitiveness has stopped to be a central determinant of export success. This variable seems like a central factor for the exports -just four industries are negatively related with this indicator (17+19,

20, 32 and 31)- which means that the scale of production gives an advantage to the industries. In particular, this dimension appears to be fundamental in the case of Motor vehicles (34) and Food, beverages, and tobacco (15+16) and in less magnitude for Other non-Metallic products (26), Basic metal (27), Pharmaceuticals (2423), Coke and refined petroleum products (23), and Aircraft and spacecraft transport equipment (353). In between the extremes of success and failure in export performance at the different ends of the map, there are intermediate industries with acceptable level of exports. Hence, product capacity and in a less extends cost, are fundamental determinants of the manufacturing exports although the dimension of the map does not seem to reflect the exporting success of all the industries. Finally, we consider the spillover effects (Figure 3g) over the exporting industries; which is surprisingly a source of competitiveness for a reduced number of industries, restricting the export performance in a aggregate level and by industry. In fact, following the results of the SOM algorithm, just Machinery and equipment (29) and Computing machinery (30) benefit from the multinationals technology. It is worth mentioning that, once again, the greater effects of these indicators do not occur in the major exporting industries.

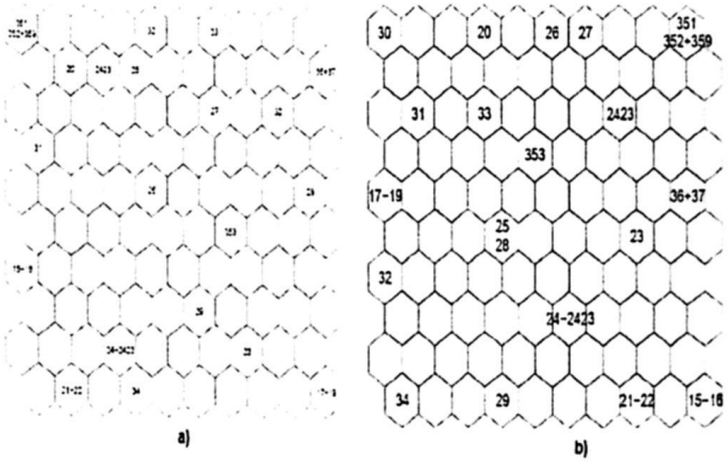


Fig. 1. SOM obtained for exports in 1985 (a), and in 2006 (b). Labels indicate the kind of exportation (see text).

4 Conclusions

In this document we have shown that the SOM can be used as an alternative and effective tool for economic analysis, in particular for cluster applications. Our results have demonstrated the efficacy of this approach in explaining the export

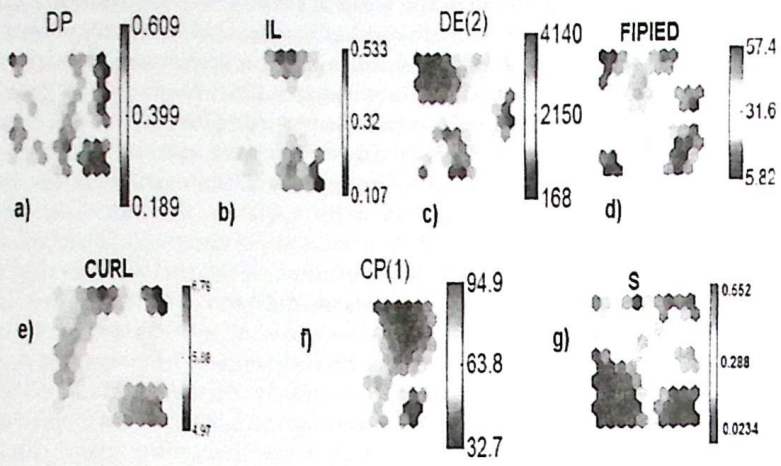


Fig. 2. SOMs for 1985. It is indicated the level of some of the variables that define the multidimensional space. DP: Product differentiation xi; IL: Lerner Index; DE(2): Foreign demand (US imports); FIPIED: Foreign direct investment; CURL: Relative Unit Labour Cost; CP(1): Domestic production capacity (product volume index); S: Technological spillovers.

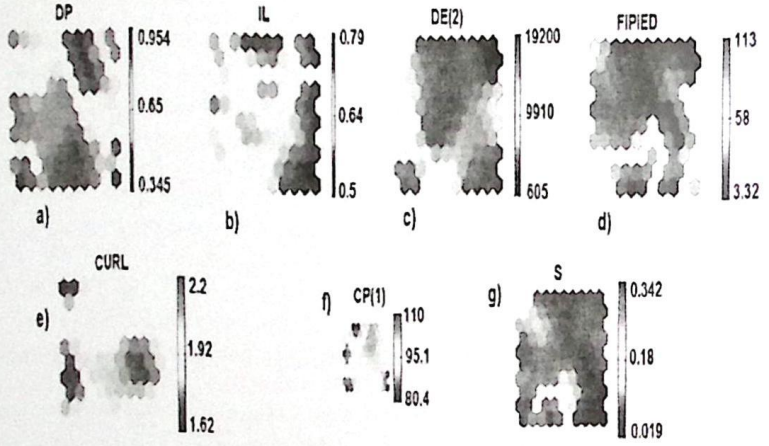


Fig. 3. SOMs for 2006. It is indicated the level of some of the variables that define the multidimensional space. DP: Product differentiation xi; IL: Lerner Index; DE(2): Foreign demand (US imports); FIPIED: Foreign direct investment; CURL: Relative Unit Labour Cost; CP(1): Domestic production capacity (product volumen index); S: Technological spillovers.

performance of manufacturing industries classified in concordance of their technological intensities. Then, based in an abundant set of variables associated with different aspects of the export process that includes technological capacities, market structure, external demand, labor cost, foreign investment, scale economies, among many others, we built a map of export performance. The set consisted of 80 indicators which described different aspects of the Mexican export success. The SOM algorithm determined the similarity between various attributes and performed the clustering of similar industries. The SOM methodology offers the economist a different perspective for the export behavior interpretation. Particularly, the SOM has great utility because it reflects two aspects of export performance of Mexican industries. First, industries are distributed along the map implying there are notable disparities among them. Principally, the export performance map is derived from a set of variables that affect in different ways -in magnitude and direction- the industries. We think this is the main reason of the dispersal distribution of the industries in the SOM. In other words, each industry has a relative export success that depends on different variables that hardly allows us to group them in defined clusters. Second, despite these differences, some industries are relatively near to each other which allow us to suggest these industries, and some of its firms, share characteristics in terms of export success. Finally, the SOM analysis finds out that in 1985 industries did not show a clear technological pattern for the considered industries. However, this situation has tended to change in 2006, implying that high and medium-high tech export industries exploit in major grade their technological capacities as a form to compete in the US market.

References

1. Kohonen, T. Self-Organizing maps. 3rd. ed. Springer-Verlag. (2000).
2. Cottrell, M. Fort, J.C., Pagés, G. Theoretical aspects of the SOM algorithm. Neurocomputing 21 (1998) 119-138.
3. Ritter. H. Self-Organizing Maps on non-euclidean Spaces Kohonen Maps, 97-108, Eds.: E. Oja and S. Kaski, (1999).
4. Vesanto, J. Som-based data visualization methods. Intelligent Data Analysis, 3(2):111126, (1999).
5. Rissi, F. Visual Data Mining and Machine Learning. ESANN'2006 proceedings - European Symposium on Artificial Neural Networks Bruges (Belgium) (2006).
6. Kaski, S., Kohonen, T. Exploratory data analysis by the self-organizing map: strcutures of welfare and poverty in the world. In Apostolos-Paul N. Refenes, Yaser Abu-Mustafa, John Moody, and Andreas Weigend (Eds.) Neural Networks in Financial Ecnonomy. Proc. of the 3rd. Int COnf. on Neural Networks and Capital Markets. World Scientific. (1996).
7. Deboeck, G. Kohonen, T. Visual Explorations in Finance with Self-Organizing Maps Springer. (1998).
8. Grossman, G.,Helpman, E. 'Technology and Trade'. CEPR Discussion Paper no. 1134. London, Centre for Economic Policy Research. (1995).

Construction of Autosimilar Electoral Units using Self-Organizing Maps

Alberto García Aguilar[1], José Carlos Méndez de la Torre[1], and Leopoldo Trueba Vázquez[1]

Universidad Autónoma de Zacatecas, Unidad Académica de Matemáticas,
Paseo a la Bufa esquina con Calzada Solidaridad,
Zacatecas, Zac., México, C.P. 98068

Abstract. In this paper we present a method of constructing autosimilar electoral units, using the Self-Organizing Maps (SOM) for detecting those electoral sections that present similar results to the complete universe. A particular case is presented that show the benefits of the application, in the capital of Zacatecas state; this method can be applied to electoral surveys and preliminar results of a given election.

1 Introduction

The obtention of satisfactory results in surveys for measure the vote intention and in the preliminary results programs is not easy. In first place, the electoral universes generally present a big cuantity of electoral units for recolecting the votes and in adition, some universes present high geographic and population dispersion. Another of the principal problems is derived of the fact that the electoral units do not have, by one side, an homogeneus behavior in the intention of the vote emission that allows using the method of standar stratified sampling.

Taking into acount various elements that are presented in the electoral units leads us to a problem with the design of the methodology that is going to be used in the selection of the units like representative units. It is usual to use the random selection acording the number of electors, however, this present the problem that sampling frame is based in the cuantity of electors and not in the historical results of the election. The statistic orthodoxy implies that the classical methods are good when sampling frame is good, and this good frame is precisely the electoral behavior and not the cuantity of people that make up the electoral units of the universe. For example, a universe with high electoral dispersion requires of selecting a big number of elements by each electoral unit and this take us to a bigger size and cost of the sample.

The principal target has to be selecting those electoral units that present a similar behavior to the real historical development of the trends of results of the vote orientation and the citizen's opinions. This is what we define like autosimilar electoral units, this denomination has its origin in the mathematical concept of autosimilarity, used in fractals and that is the property that guarantee the reproduction of the geometric structure in a different scale.

Advances in Artificial Intelligence: Algorithms and Applications
Research in Computing Science 40, 2008, pp. 39-48

An electoral unit is generally composed by one or some boxes that are the basic structure in the elections, so then one of the first steps for studing the electoral results should be analyse a set of the electoral units, but, Wich units has to be studied? It is not desirable to study all of them, a procceding for clostering the units should be done to make the study easier. In this clasification, those units that are closer or that are similar to the behaviour of the whole electoral universe are very important.

Extrapolating the mathematical concept of autosimilarity to the electoral results implies a change of the scale that can be made using the percentages obtained by the different political parties in the unit and in the total of all the units. Then, the problem is reduced to find the units or the sections that have very similar historical percentages of elections to the complete universe under study.

2 The Principle of the SOM

Acording to [1], in the documentation included of the software "SOM_PAK" and that is reproduced in this section: There exist many version of the SOM [2]. The basic philosophy, however, is very simple and already effective as such.

The SOM here defines a mapping from the input data space $\mathbb{R}^n$ onto a regular two-dimensional array of nodes. With every node i, a parametric reference vector, $m_i \in \mathbb{R}^n$ is associated. An input vector $x \in \mathbb{R}^n$ is compared with the m_i and the best match is defined as *response*: the input is thus mapped onto this location. The array and the location of the response (image of input) on it are supposed to be presented as a graphic display.

One might say that the SOM is a *nonlinear projection* of the probability density function of the high-dimensional input data onto the two-dimensional display. Let $x \in \mathbb{R}^n$ be an input data vector. It may be compared with all the m_i in any metric; in practical applications, the smallest of the Euclidean distances $||x - m_i||$ is usually made to define the best-matching node, signified by the subscript c:

$$||x - m_c|| = \min_i \{||x - m_i||\} \text{ or } c = arg\min_i \{||x - m_i||\} \tag{1}$$

Thus x is mapped onto the node c relative to the parameter values m_i. An *optimal* mapping would be one that maps the probability density function $p(x)$ in the most *faithful* fashion, trying to preserve at least the local structures of $p(x)$. (You might think of $p(x)$ as a flower that is pressed!). In the practical applications for wich such maps are intended, it may be usually self-evident from daily routines how a particular input data set ought to be interpreted. By inputting a number of typical, manually analized data set and looking where the best matches on the map according to Eq. (1) lie, the map or at least a subset of its nodes can be labeled to delineate a *coordinate system* or at least a set of characteristic reference points on it according to their manual interpretation. Since this mapping is assumed to be continuous along some hypotetical *elastic*

surface, it may be self-evident how the unknown data are interpreted by means of interpolation and extrapolation with respect to these calibrated points.

3 Autosimilar Units

The SOM is a good tool that, adapted, may allow us clasify the Electoral Units (EU) in groups and subgroups characterized by the electoral results. Their application in this case, is to group the electoral results of the EU, in some groups and subgropus that have similar characteristics. In our case is of vital importance the group of EU that give us an homogeneous group or something similar with the results of all the universe, this is finding the autosimilar EU.

To make a detailed analysis and knowing if the electoral behavior of one or some EU can be similar to the whole universe is necessary to define how the data will be treated. The results of one election is normally presented in absolutes votes and in tabular way (see Tab. 1), in wich the EU are the rows and the political parties (PP) are the columns, filling the array with the obtained votes by each PP in every EU, plus that, we add columns for the null votes and the total of electors or electoral roll.

Table 1. Typical representation of absolute electoral results.

	$\mathbf{PP}_1$	$\mathbf{PP}_2$	...	$\mathbf{PP}_n$	**Effective votation**	**Electoral Roll**
$\mathbf{UE}_1$	$V_{1,1}$	$V_{2,1}$	...	$V_{n,1}$	$T_1 = \sum V_i, 1$	P_1
$\mathbf{UE}_2$	$V_{1,2}$	$V_{2,2}$	...	$V_{n,2}$	$T_1 = \sum V_i, 2$	P_2
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\mathbf{UE}_m$	$V_{1,m}$	$V_{2,m}$	...	$V_{n,m}$	$T_1 = \sum V_i, m$	P_m
Total	$V_1 = \sum V_{1,j}$	$V_2 = \sum V_{2,j}$	...	$V_n = \sum V_{n,j}$	$T = \sum T_i$	$P = \sum P_i$

In first place, we need to prepare the electoral data in a way such that some comparations can be made, this implies using an adecuated scale for all the EU. The adecuated scale is transforming the absolut data to a relative scale for each EU, this is, to take the percentages obtained for each PP in each EU

$$v_{i,j} = \frac{V_{i,j}}{T_j} \qquad (2)$$
$$v_i = \frac{V_i}{T}$$

This give us the advantage that all the EU sums one, regardless of the sum of the votes and the cuantity of the electoral roll (see Tab. 2).

Table 2. Typical representation of relative electoral results.

	PP_1	PP_2	...	PP_n	Null	Effective	Citizen Part.
UE_1	$V_{1,1}$	$V_{2,1}$	...	$V_{n,1}$	$V_{a,1}$	1	p_1
UE_2	$V_{1,2}$	$V_{2,2}$	...	$V_{n,2}$	$V_{a,2}$	1	p_1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
UE_m	$V_{1,m}$	$V_{2,m}$	...	$V_{n,m}$	$V_{a,m}$	1	p_m
Total	$V_1 = \sum V_{1,j}$	$V_2 = \sum V_{2,j}$	...	$V_n = \sum V_{n,j}$	V_a	1	p

In this data array, the rows represent to the EU, the columns to the PP y the cross-values are the percentages obtained for the different PP that have participated in the election process. After that, it is added the total percentage as an additional row in the data array and it is considered like another unit.

4 Zacatecas Case

Before going inside in a detailed analisys of the methodology used for finding the autosimilar sections, let us see the electoral stage: in the city of Zacatecas, capital town of Zacatecas, the City Hall is renewed every 3 years, aditionally, for being soberan part of the Zacatecas state, the citizenship participates also in the election of the local legislature representation at the same time. The state goverment is renewed every 6 years, and this election takes place in a concurrent way when both periods match. In the other hand, the election of the republic's president and senators its made every 6 years and the election of federal members of parlament every 3 years, this two last elections are, in the same way, concurrent but independent of the local ones. Of the foregoing, we have that in Zacatecas 20 elections have been done in 9 times, and there are 111 electoral units in the town of Zacatecas.

With the general view of the behavior of the electoral preference for the political parties, we can ask, How the behavior at the electoral section level is?, or even more, Does one or some electoral sections that aproximate the historical behavior of all the political parties exist? If it is true, it would be better to study that or those autosimilar sections that could be used like good aproximations of posible future elections results.

In this work, we present a method for finding the autosimilar electoral sections and after that the best of them will be used for having a good approximation to the results of the 2006 federal elections.

5 Results

In this section are presented the results of aplying the SOM to the array formed using the information of electoral sections and the total (rows) with the votes obtained by the PP from the 1995 to 2000 (columns) in the Zacatecas town [3].

Different scenarios are proposed: autosimilar sections of each election from 1995 to 2000, autosimilar elections for some combinations of elections and for all the federal and local elections, finally it is compared the use of the autosimilar section with the best approximation of the results in the federal elections of 2006.

5.1 Different Scenarios

Aplying the SOM to a local electoral process that is composed by the election of the principal political parties in the election of Mayor and local members of parlament in 1995 we get that the section 1840 is the best approximation to the total percentage of the town.

Also, there exist other sections that can be used like autosimilar: 1788, 1790, 1821 and 1827. This can be observed in the central region of the Fig. 1.

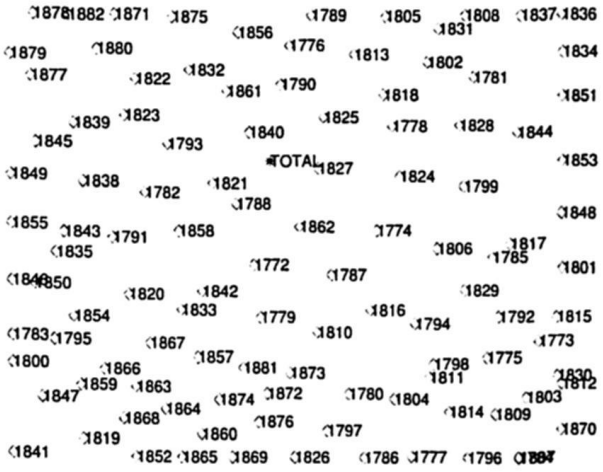


Fig. 1. SOM's result for the Mayor election and local members of parlament in 1995.

We obtain the Fig. 2 applying the SOM to the correspondent data of the election for federal members of parlament in 1997. Now, we find 2 sections with a percentage behavior very close to the whole town behavior. These are the sections 1810 and 1835. Considering those distances we can find other sections that could be proposed like autosimlar: 1804, 1820, 1838, and 1840.

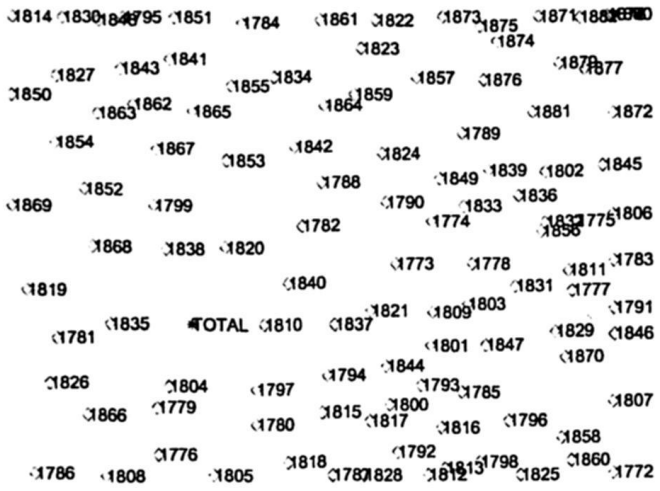


Fig. 2. SOM's result for the federal members of parlament election in 1997.

Now, let us take the 3 variables of 1998, the Mayor election, local members of parlament and governor. The result can be observed in the Fig. 3, where we can check that the best porcentual approximation section is the 1864 section, and other good approximations are: 1795, 1805 and 1844.

To conclude this part, we show the result of the SOM for the Republic's President elections and federal members of parlament in the 2000 year. In the Fig. 4 it can be analyzed that in this ocassion we can not find a section that is the best percentage approximation, those that can be used like that, are: 1780, 1822, 1854, 1863 and 1867.

This is a good point for comment that the analysis of some scenarios[4] that we take, have already been done using the cluster technic [5, 6]. The remarkable fact here, is that comparing this results with the results of the cluster analysis, similar conclusions are obtained [7]. We get the same results in the autosimilar sections, but when we look to the *other good approximations*, we realized that they are not the same, but there are coincidences.

In the Tab. 3, we have a summary that contains the election year, the best electoral section for each election and in the third column, the other good autosimilar sections.

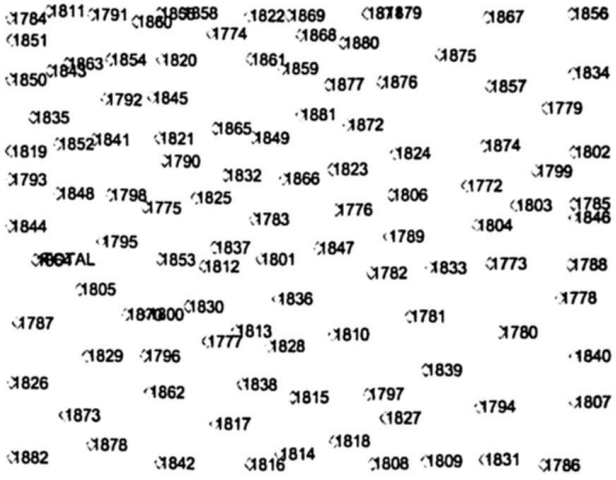


Fig. 3. SOM's result for the Mayor election, local members of parlament and governor in 1998.

Table 3. Summary of autosimilar electoral sections.

Year	Best section	Other approximations
1995	1840	1788, 1790, 1821, 1825, 1827
1997	1810, 1840	1804, 1820, 1835, 1838
1998	1864	1795, 1805, 1844
2000	1780, 1854	1822, 1863, 1867

5.2 Historic Autosimilar Section

Applying the SOM technic for the 111 sections and the town behavior, only considering the votation for the principal political parties in the whole local and federal elections since 1995 to 2001, was obtained as the best autosimilar electoral section, the section 1793, because it is the closest to the historic town behavior. This is shown in the Fig. 5.

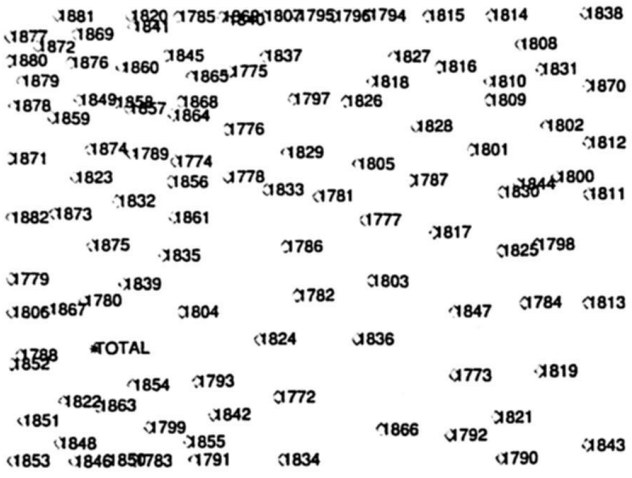


Fig. 4. SOM's result for the Republic's President election and federales members of parlament in 2000.

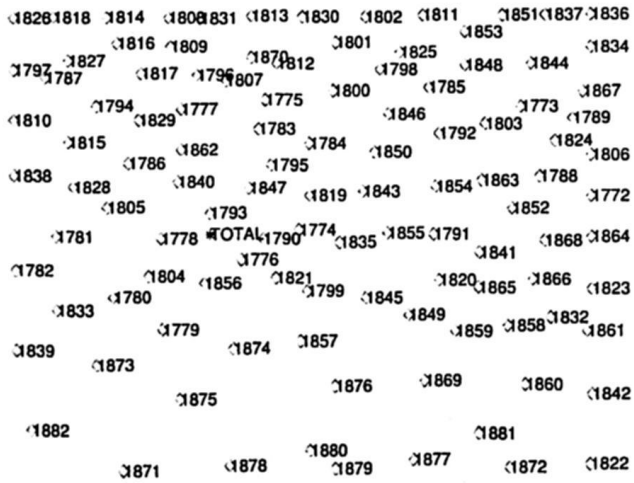


Fig. 5. SOM's result for the historic behavior considering all the elections since 1995 to 2001.

The Fig. 6 presents the historical behavior of the section 1793 as the best historically, compared with the town's historic total percentage in the different elections. In the notation, the first letter corresponds to the type of the election: M for mayor, D for members of parlament, G for governor and R for the republic's president; the las two numbers indicate the year of the election and in the middle we have the name of the political party: A for PAN, I for PRI, D for PRD, T for the PT, A*C for Alianza por el Cambio and A*M for Alianza por México.

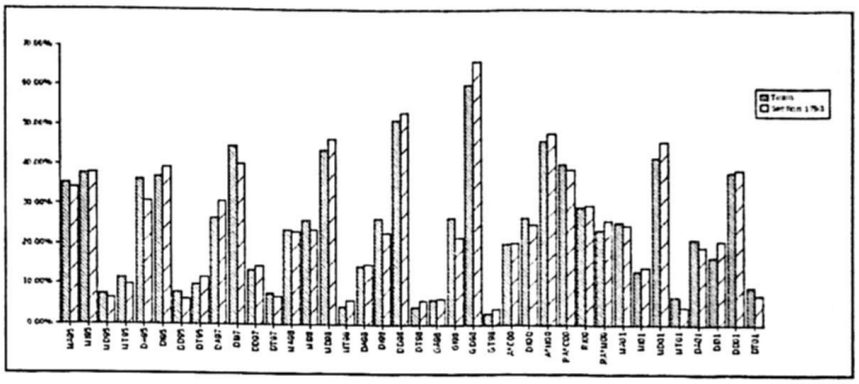


Fig. 6. Historical behavior of the section 1793 respect the town official results in the elections of 1995 to 2001.

5.3 Using Autosimilar Sections to Forecast Results

Now, we introduce the use of autosimilar sections like the best approximations for electoral processes results, it is used the best of the sections, the section 1793, taked since 1995 to 2001. Our goal is to compare this section with the town percentages obtained by the political parties that participate in the electoral process of the 2006 for President of the Republic. The summay of the analysis is shown in Tab. 4.

There, in Tab. 4 we can see that the maximum error is less than 2.5%, the half of what is commonly used in classical sampling for this kind of issues. So, we have that the autosimilar sections obtained with the SOM technic can be used to forecast results of future elections.

Table 4. Comparation of the section 1793 percentages with the results of Zacatecas town for the electoral process in the 2006 for Republic's President.

Political Party	Section 1793	Town Behavior	Error
PAN	32.7 %	35.1 %	2.4%
Alianza por México	20.2 %	17.4 %	2.8%
Coalición por el Bien de Todos	38.0%	37.2 %	0.9%
Nueva Alianza	1.1%	1.3%	0.2%
Alternativa	1.9%	4.3%	2.4%
Not registrated	3.8%	2.7%	1.1%
Nulls	2.3%	2.1%	0.1%

6 Conclusions

Through the SOM application, it was possible to find electoral sections that have an historical behavior similar to the development of local and federal electoral processes in the town of Zacatecas since 1995 to 2001.

This technique for finding autosimilar sections applying SOM, can be used not only in the Zacatecas town but can also be applied for finding similar sections in Local and Federal Electoral Districts at the State level and for the whole country. Then, the goal is to find a group of electoral sections that have a similar historical behavior to the universe of electoral sections.

The importance of having a group of autosimilar sections is to have an instrument that can be used for improve work about the surveys of voting intentions, because you can generate control surveys, and with those control surveys you can verify if you get the same results of a survey that has been designed under an statistical methodology.

The autosimilar sections can also be applied as an instrument in itself for making surveys of voting intentions, that allows to have estimators trend with lower or equal approximation errors to the classical sampling.

References

1. Kohonen, T.; Hynninen, J.; Kangas, J.; Laaksonen, J., *SOM_PAK The Self Organizing Map Program Package.* SOM Programming Team of the Helsinki University of Technology Laboratory of Computer and Information Science. (1995)
2. Kohonen, T.: Self-Organizing Maps. Springer-Verlag, Berlin (2001)
3. *Instituto Electoral de Estado de Zacatecas.* http://www.ieez.org.mx/principal.htm
4. Velasco Olvera, Hilda. ***Análisis de la votación municipal histórica en Zacatecas.*** Universidad Autnoma de Zacatecas (2002).
5. Dallas E. Johnson., ***Métodos multivariados aplicados al análisis de datos***, International Thomson Editores (1998).
6. Hair Jr. Joseph F., Anderson Rolph E., Tatham Ronald, Black William C. ***Análisis Multivariable*** 5 Ed., Prentice Hall ed. Madrid (1999).
7. Garcia A., Trueba L., Mercado G., ***Construcción de Unidades Electorales Autosimilares.*** Revista IEEM Apuntes Electorales, Año VII, Número 33, pp. 11-42 (2008).

A Novel Approach to the Analysis of Volcanic-Domain Data using Self-Organizing Maps: A Preliminary Study on the Volcano of Colima

JRG[1]Pulido, EMR[2]Michel, MA[3]Aréchiga, and G[4]Reyes

[1] Faculty of Telematics, University of Colima, México, jrgp@ucol.mx
[2] Faculty of Telematics, University of Colima, México, ramem@ucol.mx
[3] Faculty of Telematics, University of Colima, México, mandrad@ucol.mx
[4] Volcanic observatory (RESCO), University of Colima, México, gard@ucol.mx

Abstract. This paper describes an approach for helping in the enduring task of analysing volcanic-domain data. This proposal allows domain experts to have a view of the knowledge contained in and that can be extracted from the digital archive. Specific-domain ontology components with further processing, and by embedding that knowledge into the digital archive itself, can be shared with and manipulated by software agents. In particular, we deal with the issue of applying an artificial learning algorithm, Self-Organizing Maps, to volcano-tectonic signals originated by the activity of the Volcano of Colima, Mexico. By applying this algorithm we have generated clusters of volcanic activity and can readily identify situations of risk for predicting important events.

1 Introduction

Every day the activity of the different volcanoes in the world attract the attention of the goverment and scientists. This activity varies in intensity. Volcanic seismology is, in most cases, one of the most deadly natural disasters in the world. In the worst case, whole areas are devastated by erupting volcanoes, including communities living near by. A number of computacional architectures and resources have been set up all around the world to monitor, forecast, and alert people regarding volcano activity. This paper is a first approach to the problem of volcanic seismology from the computational perspective, in particular applying Self-Organizing Maps.

The next generation of volcano domain computational tools require that the huge amount of information generated by volcanoes and contained into digital archives is structured [3]. In the last few years a number of proposals on how to represent knowledge via ontology languages have paraded [8, 14, 11, 22]. Now that OWL has become an standard [20], the real challenge, in the context of the semantic web, has started. In this paper in particular, the volcanic-domain problem is addressed. Eventually, the kowledge contained into volcano activity digital archives will become semantic knowledge, ie software agents will be able to understand, manipulate, and even carry out inferencing and reasoning tasks

Advances in Artificial Intelligence: Algorithms and Applications
Research in Computing Science 40, 2008, pp. 49-59

for us. Converting such as digital archives into semantic ones is to take much longer if no semi-automatic approaches are taken into account to carry out this enterprise. This is what our paper is all about.

The remainder of this paper is organized as follows. In section 2 some key concepts on ontologies are introduced. Some related work is presented in section 3. Our approach is described in section 4. The paper concludes in section 5 with some thoughts on the approach we have applied to analyse volcanic-domain data.

2 The Purpose of Ontology

Scientists among disciplines require a framework in order to be able to interact with each other. Ontology is a framework that makes it possible for people to communicate in a consistent, complete, and distributed way. Even more, we are able to encode for a particular domain:

- entities, objects, processes, and concepts.
- relationships of entities, objects, processes, and concepts.
- relationships across discipline areas.
- domain-dependant axioms.
- multilingual knowledge of the domain.
- assumptions, parameter settings, experimental conditions as well.

These are useful forms of knowledge representation which may be used to support the design and development of intelligent software applications and expert systems. One of the most common uses of ontologies is to support the development of agent-based systems for web searching, for example those described in [23]. For this interaction to be possible, agents must share a common ontology, or at least a common wrapper to existing information structures.

In Table 1, an excerpt of the *volcano* ontology in OWL defined by the Semantic Web for Earth and Environmental[5]Terminology is presented. Some superclasses are shown. From this, a taxonomy can then be derived or vicev-ersa in a semi-automatic way by means of appropriate ontology software tools. Representing knowledge about a domain as an ontology is a challenging process which is difficult to do in a consistent and rigorous way. It is easy to lose consistency and to introduce ambiguity and confusion [2]. Ontologies can be expressed with varying degrees of formality according to the level of formalisms they can be written, however the following four categories are the most common ways to express them [38, 7]:

1. **Highly informal** written using unstructured natural language, usually as a list of terms, no axioms, no glosses at all, stored in a raw file.
2. **Semi-informal** restricted and structured using natural language, no axioms, glosses appear usually as a data-dictionary, may use more complex data structures to be stored.

[5] http://sweet.jpl.nasa.gov

Table 1. A excerpt of a volcano ontology.

```
<owl:Class rdf:about="#Volcano">
    <rdfs:subClassOf rdf:resource="#TopographicalRegion"/>
    <rdfs:subClassOf rdf:resource="#VolcanicSystem"/>
.
.
.
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:onProperty rdf:resource="#primarySubstance"/>
        <owl:someValuesFrom rdf:resource="&substance.owl;#Magma"/>
      </owl:Restriction>
    </rdfs:subClassOf>
  </owl:Class>
```

3. **Semi-formal** using a formally defined language, a collection of concepts with a partial order induced by inclusion, basic axioms, some basic searching tasks may be carried out, stored in centralized databases.
4. **Rigorously formal** including axioms, theorems and proofs, inference and reasoning tasks can be carried out, stored in distributed repositories.

The last two are the most appropriate for software agents to use in the context of the semantic web, in particular the last one as it provides mechanisms to carry out inference and reasoning.

3 Related Work

One of the most important aspects of monitoring volcano activity is forecasting, on one hand. An important number of research papers on this area are found in the literature. On the other hand, in the context of the *semantic web*, perhaps the most important aspect is related to mapping unstructured data into software agent enable knowledge [3]. In the next subsections we have a brief look at some work done on the computational aspects of volcanology. We move then onto the ontology construction and taxonomy systems aspects.

3.1 Volcanology

A vast source of research is [41]. In this book, the properties of volcano-tectonic earthquakes are described. A methodology and some applications for predicting eruptions are discussed. A classification of volcanic earthquakes is also presented. A study of volcanic explosions carried out onto four volcanos is described in [39]. This study focuses on applying several basic statistical techniques to small-scale events in trying to find clustering properties. An important software tool for volcanic-domain data is visualization. In [16] a study that explores these

techniques is presented. Researchers in the geoscience areas consider increasingly important using visualization and clustering software tools as an useful device to analyse data. The Volcano of Colima, Mxico, is one of the most active volcanoes in the world and the Telemetric Seismic Network (RESCO) monitors it. In Table 2, an excerpt of volcano signal sampling is presented.

Table 2. Seismic signal samples. Date and time omitted

#####	TipEvent	EZV4	EZV5	Lat.	Long.	Mag.	Prof.	VelAp.	#E	Archivo
00046	ve	408	416	19.519	-103.629	3.8	0.8	17.97	6	02030131.rss
00047	lp	38	46	19.528	-103.612	1.0	2.8	14.68	6	02030202.rss
00048	ve	380	385	19.525	-103.607	3.7	2.9	11.57	6	02030240.rss
00049	lp	---	25	19.831	-103.526	0.6	15.0	10.43	3	02030255.rss
00050	lp	26	26	19.815	-103.489	0.7	15.0	12.09	3	02030257.rss
00051	lp	34	24	19.826	-103.512	0.8	15.0	10.31	3	02030258.rss
00052	rf	75	---	------	-------	---	---	-----	1	02030640.rss
00053	lp	12	12	19.827	-103.516	-0.1	15.0	11.00	3	02030813.rss
00055	ve	401	410	19.525	-103.628	3.7	1.7	17.00	6	02031045.rss

3.2 Constructing Ontologies

For the volcano domain ontology construction process it is important to identify knowledge components and not to start from scratch. A good ontology assures scientists that software agents can reason properly about the domain knowledge and, for instance, forecast important events. Web ontologies can take rather different forms [36]. In [7] an early approach, the so-called *Simple HTML Ontology Extension* (SHOE) in a real world internet application is described. This approach allows authors to add semantic content to web pages, relating the context to common ontologies that provide contextual information about the domain. Most web pages with *SHOE* annotations tend to have tags that categorize concepts, therefore there is no need for complex inference rules to perform automatic classification.

Two ubiquitous and inter-related concepts in meta-level descriptions of information are *hierarchy* and *proximity*. Data samples, in a *dataset*, can be described as being *close* to one another if they are similar in some sense (eq.1). Two samples might be close in one respect, say writing style, but distant in another respect, for example content. We are more interested in the latter. On one hand, a distance measure applied to a set of samples results in a partial order relation which can form the basis for an ontology [31], for instance by using the Euclidean distance:

$$c_{ij} = \frac{\sum_k x_k y_k}{\sqrt{\sum_k x_k^2 \sum_k y_k^2}} \tag{1}$$

It is desirable to have an objective measure of the *quality* of a given ontology in order that a decision can be made as to whether one representation is better or worse than another. At this point, it is very important to state that a domain expert must be always part of the team for validating the ontology.

3.3 Taxonomy systems

Creating a volcano domain taxonomy scheme may help improve predicting software systems. Again, ontology is a useful framework to construct such a schemes. Support for browsing using classification hierarchies is an important tool for users of digital archives, eg *Yahoo*[6]categories. Users would like the data to be structured in a way that makes sense from their point of view. The purpose of browsing an environment is to present the data in a structured way such that this facilitates the discovery of information for a given purpose. We are able to do so by using ontologies as taxonomy systems as well. In [32] a distributed architecture for the extraction of meta-data from WWW documents is proposed which is particularly suited for repositories of historical publications. This information extraction system is based on semi-structured data analysis. The system output is a meta-data object containing a concise representation of the corresponding publication and its components. These meta-data objects can be classified and organized and then interchanged with other web agents. In [17] an intelligent agent for libraries is described. This inhabits a rich virtual

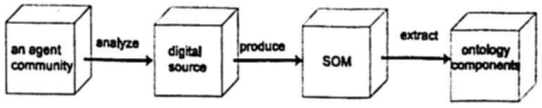


Fig. 1. Basic Approach. The ontology components from the domain are clustered together by a SOM. Further processing would allows us to embed this knowledge by means of OWL, for instance, into semantic digital archives for the current web to be transformed into software agent enable knowledge.

environment enhanced with various information tools to support searching.

Another interesting project is presented in [19], where the results of applying the *WEBSOM2*, a document organization, searching and browsing system, to a set of about 7 million electronic patent abstracts is described. In this case, a document map is presented as a series of HTML pages facilitating exploration. A specified number of best-matching points are marked with a symbol that can be used as starting points for browsing. Documents are grouped using Self-Organizing Maps (SOM), and then a graphical real-world metaphor is used to present the documents to users. That system was used as a front-end to a search engine. SOMLib and libViewer [28]. In SOMLib, maps can be integrated

[6] http://www.yahoo.com

to form a high-level library. This allows users to choose sections of a library to create personal libraries. Hierarchical feature maps consist of a number of individual self-organizing maps, and are able to represent the contents of a document archive in form of a taxonomy [24].

4 Our Approach

Our system (fig.1) can be regarded as a set of software tools that helps in the semi-automatic construction of domain-specific ontologies, in particular by clustering together a number of elements of the following sets [31]:

1. **Set of objects** (entities, concepts).
2. **Set of functions** (for example *is-a*).
3. **Set of relations** (*has* for instance).

Domain experts are always needed in order to validate the ontology components that have been identified. It can be inferred that an ontology should be produced in a *bespoke* manner to suit its purpose. This of course raises the crucial question of how such a purpose may be identified and specified. Linguistic resources such as *Wordnet* may help the domain expert in the validation of the ontology. Links to a set of *hyponyms*, incluiding instances, in *WordNet*[7] as explained in [25] can be introduced. Orology, speleology, and geophysics are hyponyms of geology. Asama, Pinatubo, and Colima are instances of Volcano for example.

4.1 Preparing the dataset

The obvious source of information for constructing a volcano-domain ontology is the data contained in the digital archives themselves. Datasets can be regarded as high dimensional vector spaces and can be represented either in a tabular form as shown in the following table:

D	v_1	$\cdots$	v_m
s_1	a_{11}	$\cdots$	a_{1m}
$\vdots$	$\vdots$	$\ddots$	$\vdots$
s_n	a_{1n}	$\cdots$	a_{nm}

or in a mathematical way as follows:

$$d_j = \sum_k a_{jk} e_k \tag{2}$$

where $\{v_1, \cdots, v_n\}$ are n-dimensional *variables*, and $\{s_1, \cdots, s_n\}$ are m-dimensional *samples*, e_k is the unit vector and a_{jk} is the frequency of occurrence of v_j in s_k.

[7] http://wordnet.princeton.edu/perl/webwn

Our system consists of two applications: Spade and Grubber [5]. The former pre-processes data and creates a dataspace suitable for training purposes. The latter is fed with the dataspace and produces knowledge[8]maps that allow us visualize ontology components contained in the digital archive. As we have mendtioned, in a semantic context, they may later be organized as a set of *Entities*, *Relations*, and *Functions*. Problem solvers use this triad for inferring new data from [9, 10, 26] and carrying out reasoning.

4.2 Visualzing ontology components

By using Self-Organizing Maps we are able to cluster together volcano-domain ontology components. SOM can be viewed as a model of unsupervised learning and an adaptive knowledge representation scheme. Adaptive means that at each iteration a unique sample is taken into account to update the weight vector of a neighbourhood of neurons [18]. Adaptation of the model vectors take place according to the following equation:

$$m_i(t+1) = m_i(t) + h_{ci}(t)[x(t) - m_i(t)] \tag{3}$$

where t $\in \mathcal{N}$ is the discrete time coordinate, $m_i \in \Re^n$ is a node, and $h_{ci}(t)$ is a neighbourhood function. The latter has a central role as it acts as a smoothing kernel defined over the lattice points and defines the stiffness of the surface to be fitted to the data points. This function may be constant for all the cells in the neighbourhood and zero elsewhere. A common neighbourhood kernel that describes a natural mapping and that is used for this purpose can be written in terms of the Gaussian function:

$$h_{ci}(t) = \alpha(t) \exp(-\frac{||r_c - r_i||^2}{2\sigma^2(t)}) \tag{4}$$

where $r_c, r_i \in \Re^2$ are the locations of the winner and a neighbouring node on the grid, $\alpha(t)$ is the learning rate ($0 \leq \alpha(t) \leq 1$), and $\sigma(t)$ is the width of the kernel. Both $\alpha(t)$ and $\sigma(t)$ decrease monotonically. The major steps of our approach are as follows:

1. Produce a *dataspace*. A dataset is created with the individual vector spaces from the domain by *spade*. In some cases, when the dataset already exists, spade carries out a pre-processing validation task.
2. Construct the SOM. A second software tool, *grubber*, is fed and trained with the dataset and ontology maps are then created .

Ontology components can be visualized clustered together from the knowledge maps created. In most cases, raw datasets have to be pre-processed. Once the dataset is a valid one, *grubber* can be fed into with them. We start with a randomly initialized map and after a training process, clusters of ontology components can be readily identified from the map. The regions on the maps are

[8] Ontology maps and knowledge maps are here used indistinctly.

formed by merging nodes that have the same most representative samples. After the maps are trained through repeated presentations of all the samples in the collection, a labelling phase is carried out. Neighbouring nodes that contain the same winning elements merge to form concept regions. The resulting maps represent areas where neighbouring elements are similar to each other. The software interface created allows us to relate information from the samples in such a way that each node has a feature that relates to its corresponding subfeatures. This can be seen as we browse the maps and that help us understand the clusters that have been formed. Some classic approaches to the problem of clustering, on one hand, include partitional methods [29], hierarchical agglomerative clustering [34], and unsupervised bayesian clustering [27]. A widely used partitional procedure is the k-means algorithm [15]. A problem with this procedure is the selection of k a priori. PCA, on the other hand, is an excellent tool for reducing the size of the dataset. It allows the distance between samples to be measured in a well-defined and consistent manner [6]. An alternative to these methods is SOM which does not make any assumptions about the number of clusters a priori, the probability distributions of the variables, or the independence between variables. A comparative of these methods is not presented here. Preliminary results were surprisingly close to our intuitive expectations. After this, some other ontology tools such as editors can be used to organize this knowledge. Then, it can be embedded into the digital archive where it was extracted from by means of any of the ontology languages that exist (fig.2). Some results of applying our approach in other domains have been reported [35], and we are now further researching on the volcano domain in order to validate our results. At this stage we have already considered the use of hybrid systems that in combination of our approach will help in the semi-automatic construction of specific-domain ontologies [21].

5 Conclusions

The vast amount of data generated by volcanoes has eventually to be transformed into semantic data. In the context of the semantic web, by using semantic knowledge, software agents are able to carry out inference and reasoning tasks for us. In the volcano-domain, software agents may be of help in forecasting important events. An ontology is a form of knowledge representation that provides a common vocabulary of concepts and relationships which may be used to inform a viewer, a search engine or to inform other software entities. Agents have to interact with other agents using these dissimilar concepts. Therefore mechanisms and forms to exchange information and knowledge among different disciplines are needed. As we have already seen, ontologies can be used to give a sense of order to unstructured digital sources such as volcano-domain data. The acquisition and representation of knowledge needs to take into account the complexity that is often present in domains as well as the needs of the agents carrying out the search, a volcano-domain expert is always needed in order to assure the quality of the ontology created.

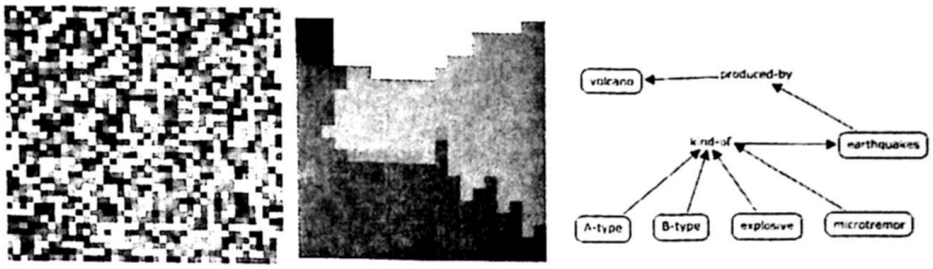


Fig. 2. After a training process, a randomly initialized SOM (left) becomes a categorized one (middle). Then an ontology can be derived (right).

In this paper we have presented a novel approach that generates clusters of volcanic activity and can readily help us identify situations of risk for predicting important events. However, some more research is required in order to fine tune the semi-automatic specific-domain ontology creation process.

References

1. K Bontcheva. Generating tailored textual summaries from ontologies. In A Pérez and J Euzenat, editors, *The Semantic Web Research and Applications*, volume 3532 of *LNCS*, pages 531–545–. Springer, 2005.
2. R Brachman. What is-a and isn't: An analysis of taxonomic links in semantic networks. *IEEE Computer*, 16(10):10–36, 1983.
3. L Crow and N Shadbolt. Extracting focused knowledge from the Semantic Web. *Int.J.Human-Computer Studies*, 54:155–184, 2001.
4. A Duineveld et al. Wondertools? a comparative study of ontological engineering tools. *Int.J.Human-Computer Studies*, 52:1111–1133, 2000.
5. D Elliman and *JRG Pulido*. Visualizing ontology components through self-organizing maps. In D Williams, editor, *6th International Conference on Information Visualization (IV02), London, UK*, pages 434–438. IEEE Computer Soc.Press, Los Alamitos, 2002.
6. G Foody. Applications of the self-organising feature map neural network in community data analysis. *Ecological Modelling*, 120:97–107, 1999.
7. A Gangemi et al. Ontology integration: Experiences with medical terminologies. In N Guarino, editor, *Formal Ontology in Info.Systems*, volume 46, pages 163–178. IOS Press, Amsterdam, 1998.
8. A Gómez and Oscar Corcho. Ontology languages for the Semantic Web. *IEEE Intelligent Systems*, 2002.
9. A Gómez et al. Knowledge maps: An essential technique for conceptualisation. *Data & Knowledge Engineering*, 33:169–190, 2000.
10. J Gordon. Creating knowledge maps by exploiting dependent relationships. *Knowledge-Based Systems*, pages 71–79, 2000.
11. J Heflin et al. Applying ontology to the web: A case study. *Engineering Applications of Bio-Inspired Artificial Neural Networks*, 1607, 1999.

12. J Hendler and E Feigenbaum. Knowledge is power: The Semantic Web vision. In N Zhonh et al., editors, *Web intelligence: Research and development*, volume 2198 of *LNAI*, pages 18–29. Springer-Verlag, Berlin, 2001.
13. V Hodge and J Austin. Hierarchical word clustering – automatic thesaurus generation. *Neurocomputing*, 48:819–846, 2002.
14. I Horrocks et al. From SHIQ and RDF to OWL: The making of a web ontology language. *Journal of web semantics*, 1(1):7–26, 2003.
15. R Johnson and D Wichern. *Applied Multivariate Statistical Analysis.* Prentice Hall, New Jersey, 4th edition, 1998.
16. B Kadlec et al. Visualization and analysis of multi-terabyte geophysical datasets in an interactive setting with remote webcam capabilities. In X Yin et al., editors, *Computational earthquake physics: simulations, analysis and infrastructure PART II.*, pages 2455–2465. Birkhauser-Verlag, Basel, 2006.
17. D Kirsh. Designing virtual libraries to help users find what they want. In C Landauer and K Bellman, editors, *The Virtual Worlds and Simulation Conf.-VWSIM'98*, pages 221–224. Soc.Computer Simulation Int, San Diego, 1998.
18. T Kohonen. *Self-Organizing Maps.* Information Sciences Series. Springer-Verlag, Berlin, 3rd edition, 2001.
19. T Kohonen et al. Self organization of a massive text document collection. In E Oja and S Kaski, editors, *Kohonen Maps*, pages 171–182. Elsevier Sci, Amsterdam, 1999.
20. L Lacy. *OWL: Representing Information Using the Web Ontology Language.* Trafford Publishing, USA, 2005.
21. S Legrand and *JRG Pulido.* A hybrid approach to word sense disambiguation: Neural clustering with class labeling. In P Buitelaar et al., editors, *Workshop on knowledge discovery and ontologies, 15th European Conference on Machine Learning (ECML), Pisa, Italy*, pages 127–132, September 2004.
22. P Martin and P Eklund. Embedding knowledge in web documents. *Computer Networks*, 31:1403–1419, 1999.
23. J McCormack and B Wohlschlaeger. Harnessing agent technologies for data mining and knowledge discovery. In *Data Mining and Knowledge Discovery: Theory, Tools and Technology II*, volume 4057, pages 393–400, 2000.
24. D Merkl. Document classification with self-organizing map. In E Oja and S Kaski, editors, *Kohonen Maps*, pages 183–192. Elsevier Sci, Amsterdam, 1999.
25. G Miller et al. Introduction to WordNet: An on-line lexical database. *International Journal of Lexicography*, 3(4):235–244, 1991.
26. E Motta et al. Ontology-driven document enrichment: principles, tools and applications. *Int.J.Human-Computer Studies*, 52:1071–1109, 2000.
27. J Principe. *Neural and Adaptive Systems, Fundamentals through Simulations*, chapter 7. Wiley, USA, 2000.
28. A Rauber and D Merkl. The SOMLib digital library system. *LNCS*, 1696:323–342, 1999.
29. B Ripley. *Pattern Recognition and Neural Networks*, chapter 1,9. University Press, Cambridge, 1996.
30. H Ritter and T Kohonen. Self-organizing semantic maps. *Biological Cybernetics*, 61:241–254, 1989.
31. G Salton. *Automatic Information Organization and Retrieval.* McGraw-Hill, New York, 1968.
32. I Sanz et al. Gathering metadata from web-based repositories of historical publications. In A Tjoa and R Wagner, editors, *9th Int. Workshop on Database and*

Expert Systems Apps, pages 473–478. IEEE Computer Soc.Press, Los Alamitos, 1998.

33. F Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47, 2002.
34. H Tanaka et al. An efficient document clustering algorithm and its application to a document browser. *Information Processing and Management*, 35:541–557, 1999.
35. *JRG Pulido* et al. Identifying ontology components from digital archives for the semantic web. In *IASTED Advances in Computer Science and Technology (ACST)*, pages 1–6, 2006. CD edition.
36. *JRG Pulido* et al. Ontology languages for the semantic web: A never completely updated review. *Knowledge-Based Systems*, Elsevier volume 19, issue 7:489–497, 2006.
37. *JRG Pulido* et al. Artificial learning approaches for the next generation web: part I. *Ingeniería Investigación y Tecnología, UNAM (CONACyT), México*, 9(1):67–76, 2008.
38. M Uschold and M Gruninger. Ontologies: Principles, methods, and applications. *Knowledge Engineering Review*, 11(2):93–155, 1996.
39. N Varley et al. Applying statistical analysis to understand the dynamics of volcano explosions. In H Mader et al., editors, *Statistics in volcanology.*, pages 57–76. Geological society for IAVCEI, London, 2006.
40. Y Yang et al. A study of approaches to hypertext categorization. *J.Intelligent Information Systems*, 18(2/3):219–241, 2002.
41. V Zobin. *Introduction to volcanic seismology.* Elsevier, Amsterdam, 2003.

Graph Matching and Pattern Recognition

(with Jixin Ma)

Shape Decomposition for Graph Representation

Bai Xiao and Peter M. Hall

Media Technology Research Center
Department of Computer Science
University of Bath, U.K.

Abstract. The problem of shape analysis has played an important role in the area of image analysis, computer vision and pattern recognition. In this paper, we present a new method for shape decomposition. The proposed method is based on a refined morphological shape decomposition process. We provide two more analysis for morphological shape decomposition. The first step is scale invariant analysis. We use a scale hierarchy structure to find the invariant parts in all different scale level. The second step is noise deletion. We use graph energy analysis to delete the parts which have minor contribution to the average graph energy. Our methods can solve two problems for morphological decomposition – scale invariant and noise. The refined decomposed shape can then be used to construct a graph structure. We experiment our method on shape analysis.

1 Introduction

Shape analysis is a fundamental issue in computer vision and pattern recognition. The importance of shape information relies that it usually contains perceptual information, and thus can be used for high level vision and recognition process. There has already many methods for shape analysis. The first part methods can be described as statistical modeling [4] [12][9] [11]. Here a well established route to construct a pattern space for the data–shapes is to use principal components analysis. This commences by encoding the image data or shape landmarks as a fixed length long vector. The data is then projected into a low-dimensional space by projecting the long vectors onto the leading eigenvectors of the sample covariance matrix. This approach has been proved to be particularly effective, especially for face data and medical images, and has lead to the development of more sophisticated analysis methods capable of dealing with quite complex pattern spaces. However, these methods can't decompose the shapes into parts and can't incorporate high level information from shape. Another problem which may hinder the application of these method is that the encoded shape vectors must be same length which need large human interaction pre-processing.

Another popular way to handle the shape information is to extract the shape skeleton. The idea is to evolve the boundary of an object to a canonical skeletal form using the reaction-diffusion equation. The skeleton represents the singularities in the curve evolution, where inward moving boundaries collide. With the

Advances in Artificial Intelligence: Algorithms and Applications
Research in Computing Science 40, 2008, pp. 63-72

skeleton to hand, then the next step is to devise ways of using it to characterize the shape of the original object boundary. By labeling points on the skeleton using so-called shock-labels, the skeletons can then be abstracted as trees in which the level in the tree is determined by their time of formation[15, 8]. The later the time of formation, and hence their proximity to the center of the shape, the higher the shock in the hierarchy. The shock tree extraction process has been further improved by Torsello and Hancock[16] recently. The new method allows us to distinguish between the main skeletal structure and its ligatures which may be the result of local shape irregularities or noise. Recently, Bai, Latecki and Liu [1] introduced a new skeleton pruning method based on contour partition. The shape contour is obtained by Discrete Curve Evolution [10]. The main idea is to remove all skeleton points whose generating points all lie on the same contour segment. The extracted shape skeleton by using this method can better reflect the origin shape structure.

The previous two shape analysis methods, statistical modeling and shape skeletonization, can be used for shape recognition by combining graph based methods. For example, Luo, Wilson and Hancock [2] show how to construct a linear deformable model for graph structure by performing PCA(Principal Component Analysis) on the vectorised adjacency matrix. The proposed method delivers convincing pattern spaces for graphs extracted from relatively simple images. Bai, Wilson and Hancock[18] has further developed this method by incorporating heat kernel based graph embedding methods. These method can be used for object clustering, motion tracking and image matching. For the shape skeleton methods, the common way is to transform the shape skeleton into a tree representation. The difference between two shape skeletons can be calculated through the edit distance between two shock trees[16].

Graph structure is an important data structure since it can be used to represent the high level vision representation. In our previous work [19], we have introduced an image classifier which can be used to classify image object on different depictions. In that paper, we have introduced an iterative hierarchy image processing which can decompose the object into meaningful parts and hence can be used for graph based representation for recognition.

In this paper, we will introduce a new shape decomposition method. Our method is based on morphological shape decomposition which can decompose the binary shapes through iterative erosion and dilation process. The decomposed parts can then be used to construct a graph structure i.e. each part is a node and the edge relation reflect the relationship between parts, for graph based shape analysis. However, morphological shape decomposition has two shortcomings. First, the decomposition is not scale invariant. When we change the scale level for the same binary shape the decomposition is different. Second, the decomposed parts contains too much noise or unimportant parts. When we use graph based methods for shape analysis these two problems will certainly produce bad influence for our results. Our new method provide two more analysis for morphological decomposition. We first solve the scale invariant problem. We decompose the shape through a hierarchy way. From top to bottom each level

representing a different scale size for the same binary shape from small to big. We decompose each level through morphological decomposition and then find the corresponding parts through all levels. We call these parts invariant in all scale levels and use them to represent the binary shapes. The second step is used to delete the noise parts which are normally unimportant and small. We construct the graph structure for the decomposed parts and use graph energy method to analysis the structure. We find the parts(nodes) which has minor or none contribution to the average energy for the whole graph structure. The rest parts are kept as important structure for the shape.

In Section 2, we first review some preliminary shape analysis operations i.e. the tradition morphological shape decomposition. In Section 3, we describe a scale invariant shape parts extraction method. In Section 4, we will describe our graph energy based noise deletion and in Section 5 we provide some experiment results. Finally, in Section 6, we give conclusion and future work.

2 Background on Morphological Shape Decomposition

In this section, we introduce some background on shape morphology operation. Morphological Shape Decomposition (MSD)[14] is used to decompose the shape by the union of all the certain disks contained in the shape. For a common binary shape image, it contains two kinds of elements "0"s and "1"s, where "0" represents backgrounds and "1" represents the shape information. The basic idea of morphology in mathematics can be described as below

$$(M)_u = \; m + u \; m \in M \tag{1}$$

. There are two basic morphological operations, the dilation of M by S and the erosion of M by S, which are defined as follows:

$$M \oplus S = \bigcup_{s \in S} (M)_S \tag{2}$$

and

$$M \ominus S = \bigcup_{s \in S} (M)_{-S} \tag{3}$$

. There are also two fundamental morphological operation based on dilation and erosion operations, namely the opening of M by $S(M \circ S)$ and closing of M by $S(M \bullet S)$. The definitions are given below:

$$M \circ S = (M \ominus S) \oplus S \tag{4}$$

$$M \bullet S = (M \oplus S) \ominus S \tag{5}$$

A binary shape M can be represented as a union of certain disks contained in M

$$M = \bigcup_{i=0}^{N} L_i \oplus iB \tag{6}$$

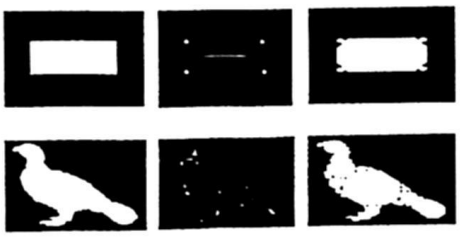


Fig. 1. An example for morphological shape decomposition.

where $L_N = X \ominus NB$ and

$$L_i = (M(\bigcup_{j=i+1}^{N})) \ominus iB \quad 0 \leq i < N \tag{7}$$

N is the largest integer which satisfy

$$M \ominus NB \neq \emptyset$$

it can be computed by an iterative shape erosion program. B is defined as morphological disks. We call L_i loci and i as corresponding radii. We follow the work by Pitas and Venetsanopoulos [14] to compute the L_i and i. This can give us an initial shape decomposition.

An example is shown in Figure1. Here two shapes(the left column) are given, in which a rectangular shape can be decomposed into five parts. In the upper-middle column of Figure 1 there are one center part and four corners. However, different with the normal shape representation which contains two elements, 0s and 1s, the loci part is represented by the elements of i and the backgrounds are still 0. It is called "Blum Ribbon". With this representation at hand, we can reconstruct the origin shape[14]. The right column in this figure shows the reconstructed shapes by using the "Morphological Ribbon".

3 Scale Invariant Structure Extraction

In the introduction part, we have emphasized the importance of incorporating graph structure representation with shape analysis. It is normal to construct a graph structure from morphological shape decomposition. We can simply treat each part as a node in the graph and the edge relationship is deduced from the adjacency between each pair of parts. If two parts are adjacent or overlap then the weight between the two corresponding nodes are non-zero. In this paper, we dilate the parts with the disk radius size two more than the origin eroded

Fig. 2. Graph structure example from morphological shape decomposition.

skeleton. For example, if two parts I and J's radius are r_i and r_j with I and J the corresponding loci, we first dilate these two parts by the radius $r_i + 2$ and $r_j + 2$. Then the weight between parts I and J is $and(I \ominus (r_i + 2)\ J \oplus (r_j + 2))/or(I \oplus (r_i + 2)\ J \oplus (r_j + 2))$ which reflect both the overlap and adjacent relationship. We can use a five nodes graph to represent the rectangular shape 2 while the center is connected with four corners.

However, the graph structure constructed from this morphological based shape decomposition method is not suitable for graph based shape analysis. It is sensitive to scaling, rotation and noise [7]. An example is shown in Figure 3 here we decompose a set of different size rectangular , we can observe two things 1) It doesn't satisfy scale invariant. As we can see, when the scale is different the decomposition results is different. At the small scale level, the rectangular shape decomposed skeleton include one line and four small triangles. While at large scale level, the skeleton include one line and twelve triangles.

3.1 Hierarchy Morphological Decomposition

We propose a solution which is to decompose the shape in different scale and find the corresponding matching parts to represent the shape. The idea is when a shape is given, we squeeze and enlarge the shape image in a sequence list. We decompose this sequence image shapes. We then find the corresponding parts for this sequence shape decomposition. The stable scale invariant shape decomposition is then found by choose the parts which appear in all different scale levels.

In Figure 3, we still use the example of the rectangular, we first squeeze and enlarge the shape by 15 percent each time. We choose three squeezed and three enlarged shapes – altogether we have five shapes. We then decompose this sequence through morphological decomposition described in the previous section. We then find the correspondence in a hierarchy style. From the correspondence results, we notice that the parts which appear in all levels are the center line and four dots in the corners. The proposed methods can solve the scale invariants

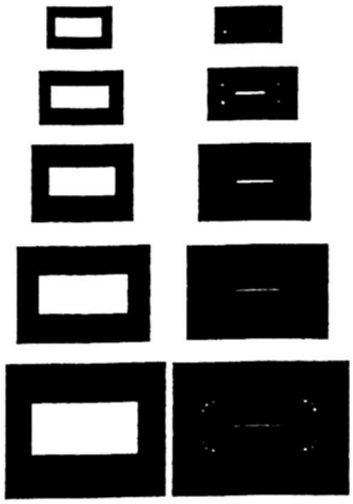


Fig. 3. Example for the same shape morphological decomposition in different scale.

problem for shape decomposition. Like SIFT feature [5], we consider the shape decomposition through a hierarchy way.

4 Graph Energy based Noise Deletion

We continue to use the idea from spectral graph theory [3] to delete the noise in morphological shape decomposition. Our idea is to use graph Laplacian energy which reflect the connectiveness and regularity for the graph to delete the parts(nodes) which has minor or none contribution to the average graph Laplacian energy per node. The solution is to iteratively delete the parts and finally stop this process when the average graph Laplacian energy per node never rise.

We first review the graph Laplacian energy [6]. The Laplacian matrix is defined as $L = D - A$, in which D is a degree matrix, and A an adjacency matrix. Laplacian graph energy has the following standard definition: for a general graph $G = (V\ A)$, with arc weights $w(i\ j)$ the Laplacian energy is

$$\mathcal{E}(G) = \sum_{i=1}^{|V|} \left| \lambda_i - 2\frac{m}{V} \right| \tag{8}$$

In which: the λ_i are eigenvalues of the Laplacian matrix; m is the sum of the arc weights over the whole graph, or is half the number of edges in an unweighted

graph; V is the number of nodes in graph. Note that $2m/\ V$ is just the average (weighted) degree of a node. Now, the Laplacian energy of a graph can rise or fall; our tests show that this rise and fall is strongly correlated with the variance in the degree matrix D. This means local minima tend to occur when the graph is regular.

Since we want to use graph Laplaican energy, we need to first construct a graph structure for morphological decomposed parts. The graph structure can be constructed through the method from previous section. We treat each parts from morphology decomposition as a node in the graph G, the edge relationship is found through the adjacency and overlap relationship between each pair of parts.

The process of noise deletion is listed below: 1) We compute the initial average graph energy for the initial state decomposition $\mathcal{E}(G)/\ N$. 2) For each iteration, we go through all the nodes in the graph G. For each node we judge whether we should delete this node. We just compare the previous average graph energy $\mathcal{E}(G)/\ N$ with the average graph energy with this node deleted $\mathcal{E}(G_{di})/\ N - i$, where G_{di} is the graph with ith nodes deleted. If the the average graph energy $\mathcal{E}(G_{di})/\ N - i$ is larger than the previous average energy then we should delete this node and update the graph structure G. 3) Repeat step two, until $\mathcal{E}(G_di)/\ N - i$ never rise. 4) Output the final decomposition.

The previous process can detect the nodes which has weak link with rest nodes in the graph. It will prune the graph structure until it near or reach regular while keep strong connectiveness within the rest nodes.

5 Experiment

In this section, we provide some experiment results for our methods. Our process can be simply described as below:

- For a given binary shape, we first squeeze and enlarge it to construct the hierarchy scale levels from small to big.
- Perform morphological shape decomposition for each level, in this paper we use Pitas and Venetsanopoulos [14] method. Find the corresponding matching parts through all levels. These parts input for the next step.
- Use the output from last step to construct the graph structure. Use average graph energy method to delete the noise nodes(parts) in the graph. Repeat this step until the average graph energy never rise. Output the final graph structure.

We experiment on shock graph database which composed of 150 silhouettes of 10 kinds of objects [16]. An example of database is shown in Figure 4.

In Figure 5, we give some results for our methods, here in the left column is the origin shape, the middle column is the pruned skeleton parts from morphological shape decomposition and the right column is the re-constructed shape by using the skeleton centers in the middle column. From this example, we can see that our algorithm can reduce some noise parts from the origin morphological

Fig. 4. Sample views of the silhouette objects

decomposition while keep the important parts. It can be seen that the reconstructed shapes are quite similar to the original shapes and thus keeps the most important information for further analysis.

In table 1 we listed the variation for the number of parts within the same class for tradition morphological shape decomposition method(MSD) and our method. It is clear that the variations for the number of parts for the tradition morphological shape decomposition is higher than our method.

Table 1. Variation for the number of parts with different shape decomposition methods.

Class Name	MSD	Our Method
Car	8.5	4.1
Children	11.4	6.7
Key	9.0	5.0
Bone	8.5	4.7
Hammer	4.5	3.2

6 Discussions and Conclusions

In this paper, we proposed a new shape decomposition method which extended the morphological methods. It can conquer two problems for the current morphological methods, scale invariant and noise. We have proposed a graph Laplacian energy based hierarchy shape decomposition. We can extract more stable graph structure by using our methods. Our next step is to use these graph structures to do shape analysis. One possible way is to combine the spectral graph invariants [17] for shape recognition. Recently, Trinh and Kimia [13] has proposed a graph generative for shape through the analysis of shock graphs. We can also extend our methods with graph generative model for morphological decomposition.

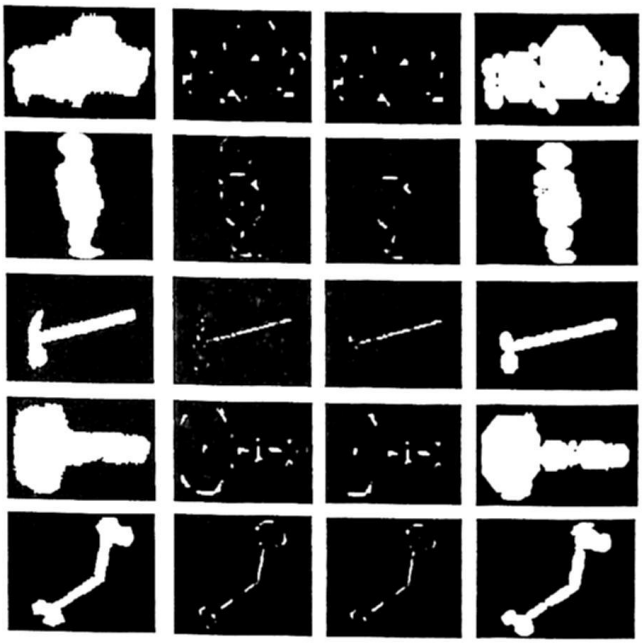


Fig. 5. Example for our methods

References

1. X. Bai, L.J.Latecki, and W.Y.Liu. Skeleton pruning by contour partitioning with discrete curve evolution. IEEE Trans. PAMI, 29(3):449–462, 2007.
2. B.Luo, R.C.Wilson, and E.R.Hancock. A spectral approach to learning structural variations in graphs. Pattern Recognition, 39:1188–1198, 2006.
3. F. R. K. Chung. *Spectral graph theory.* American Mathematical Society, 1997.
4. T.F. Cootes, G.J. Edwards, and C.J. Taylor. Active appearance models. In *European Conference on Computer Vision*, volume 1407 of *Lecture Notes in Computer Science*, pages 484–, Freiburg, Germany, 1998. Springer-Verlag.
5. D.Lowe. Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision, 1:91–110, 2004.
6. I. Gutman and B. Zhou. Laplacian energy of a graph. *Linear Algebra and its Applications*, 41:29–37, 2006.
7. Duck Hoon Kim, Il Dong Yun, and Sang Uk Lee. A new shape decomposition scheme for graph-based representation. *Pattern Recognition*, 38(5):673–689, 2005.
8. B.B. Kimia, A.R. Tannenbaum, and S.W.Zucker. Shapes, shocks, and deforamtions. Int. J. Computer Vision, 15:189–224, 1995.
9. Klassen, A. Srivastava, W. Mio, and S. H. Joshi. Analysis of planar shapes using geodesic paths on shape spaces. IEEE Transactions on Pattern Analysis and Machine Intelligence, 26:372–383, 2004.
10. L.J. Latecki and R. Lakamper. Convexity rule for shape decomposition based on discrete contour evolution. Computer Vision and Image Understanding, 77:441–454, 1999.
11. C. G. Lee and C. G. Small. Multidimensional scaling of simplex shapes. *Pattern Recognition*, 32:1601–1613, 1999.
12. H. Murase and S. K. Nayar. Illumination planning for object recognition using parametric eigenspaces. IEEE Transactions on Pattern Analysis and Machine Intelligence, 16:1219–1227, 1994.
13. N.Trinh and B.B. Kimia. A symmetry-based generative model for shape. International Conference on Computer Vision, 2007.
14. Ioannis Pitas and Anastasios N. Venetsanopoulos. Morphological shape decomposition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 12(1):38–45, 1990.
15. A. Shokoufandeh, S. Dickinson, K. Siddiqi, and S. Zucker. Indexing using a spectral encoding of topological structure. *International Conference on Computer Vision and Pattern Recognition*, pages 491–497, 1999.
16. A. Torsello and E. R. Hancock. A skeletal measure of 2d shape similarity. Computer Vision and Image Understanding, 95(1):1–29, 2004.
17. Bai Xiao and Edwin R. Hancock. Clustering shapes using heat content invariants. pages 1169–1172, 2005.
18. Bai Xiao and Edwin R. Hancock. A spectral generative model for graph structure. In *SSPR/SPR*, pages 173–181, 2006.
19. Bai Xiao, Yi-Zhe Song, and Peter M. Hall. Learning object classes from structure. In *British Machine Vision Conference*, volume 1407, pages 207–217, Warwich, 2007.

A Critical Examination of Node-Similarity Based Graph Matching Algorithms

Guoxing Zhao[1], Miltos Petridis[1], Grigori Sidorov[2] and Jixin Ma[1]

[1]School of Computing and Mathematical Sciences,
the University of Greenwich, U.K.
[2]Center for Research in Computer Science,
National Polytechnic Institute, Mexico
j.ma@gre.ac.uk

Abstract. In this paper, we shall critically examine a special class of graph matching algorithms that follow the approach of node-similarity measurement. A high-level algorithm framework, namely node-similarity graph matching framework (NSGM framework), is proposed, from which, many existing graph matching algorithms can be subsumed, including the eigen-decomposition method of Umeyama, the polynomial-transformation method of Almohamad, the hubs and authorities method of Kleinberg, and the kronecker product successive projection methods of Wyk, etc. In addition, improved algorithms can be developed from the NSGM framework with respects to the corresponding results in graph theory. As the observation, it is pointed out that, in general, any algorithm which can be subsumed from NSGM framework fails to work well for graphs with non-trivial auto-isomorphism structure.

Keywords: Graph matching, node-similarity.

1 Introduction

Graph, as a powerful and versatile mathematical tool, is widely used for the description of structural objects in many application areas such as case-based reasoning, semantic networks, document processing, image analysis, biometric identification, computer vision, video analysis, and so on.

In applications such as pattern recognition and computer vision, object similarity becomes the most important issue and based on the graph representation, object similarity is simply transfer into the similarity degree between two graphs which is known as the graph matching problems.

Various algorithms for graph matching problems have been developed, which, according to [7], can be classified into two categories: (1) search-based

Advances in Artificial Intelligence: Algorithms and Applications
Research in Computing Science 40, 2008, pp. 73-82

methods which rely on possible and impossible pairings between vertices; and (2) optimization-based methods which formulate the graph matching problem as an optimization problem. Generally speaking, on one hand, search-based methods will find optimal solutions, but require exponential time in the worst case. On the other hand, optimization-based methods normally require only polynomial-bounded computational time, but in some cases may fail to find the optimal solution.

Most search-based approaches use the idea of heuristics [13, 15, 16] to reduce the size of the searching space, while optimization-based methods have followed distinct approaches, which again can be roughly classified in to two groups, (1) traditional optimization based methods including linear programming methods [2], quadratic programming approaches [12], Bayesian methods [6], relaxation labeling [9], neural network [14], genetic algorithm [11] and so on; (2) special theory based methods including symmetric polynomials transformation [1], kronecker product successive projection [4], eigen-decomposition method [17, 18], spectral embedding approach [3], and hubs and authorities method [5, 10], etc. In general, the traditional optimization based methods transfer the graph matching problem to a classical optimization problem and traditional optimization algorithm can be directly applied, but this transformation sometimes conceals the essence and makes the algorithm hard to be analyzed and improved. On the other hand, every special theory based method usually provides a simple theoretical foundation to deduce and analyze the corresponding graph matching algorithm, so the principles and applied scope is relatively clearer. However, most special theory based methods are only applicable for very special kind of graph pairs.

In this paper, instead of a new graph matching algorithm, an abstract algorithm template called node-similarity graph matching algorithm template shall be presented. This algorithm template can be seen as an abstraction of many graph matching algorithms, such as SPGM of Almohamad [1], EDGM of Umeyama [17], HAGM of Kleinberg [10], LSKPGM of Wyk [4], etc. In this sense, a unified analysis and comparison can be provided on the starting points, execution processes and constraints of these graph matching algorithms. The rest of this paper is organized as following: Basic notations of graph matching problem and node-similarity matching algorithm template are introduced in section 2. Several concrete node-similarity algorithms are proposed, tested and compared in section 3. In section 4 we shall extends the node-similarity graph matching algorithm template for matching multiple-weighted graph pairs and in section 5 an important limitation of node-similarity graph matching algorithms are pointed out that all these algorithms are not suitable for self-similar graphs. Section 6 simply concludes this paper.

2 Graph Matching Problems

Following the definition of [1], a weighted graph G is denoted as an ordered pair (N, w), where N is a set of nodes and w is a weighting function assigning a weight $w(v_i, v_j)$ to each pair of nodes (v_i, v_j) (edge of the graph). The adjacency matrix of a weighted graph G = (V, w) is defined as $A_G = \{g_{ij}\}$, where $g_{ij} = w(v_i, v_j)$, i, j = 1, 2, ..., n, and n is the number of nodes in graph G.

Note: in this paper, only fully weighted graphs with no multiple edges are considered.

For the reason of simplifying repression, without confusion, we shall not distinguish a weighted graph G and its corresponding adjacency matrix A_G. In other words, we shall simply express the adjacency matrix of G as graph G itself.

The problem of matching two weighted graphs G and H of n nodes is to find a one-to-one correspondence between the two corresponding sets of nodes that minimizes the distance between G and H, which can be formulated in terms of Frobenius-norm (denoted as $\|\bullet\|_F$) as:

$$\arg\min_{P\in perm(n)} \|PGP^T - H\|_F \tag{1}$$

Where G and H are the adjacent matrices of the weighted graphs to be matched and perm(n) is the set of all n-by-n permutation matrices.

Note: in this paper, we only handle the matching of two graphs with the same size.

3 Node-similarity Graph Matching Algorithm Template

In general, completely solving formula (1) is nearly impossible, so we propose the node-similarity based algorithm template as follow:

$$\arg\min_{P\in perm(n)} \|P - S(G,H)\|_F \tag{2}$$

Where S is a node-similarity function from $R^{n\times n} \times R^{n\times n}$ to $R^{n\times n}$ satisfying

$$S(PGP^T, H) = S(G,H)P^T, \text{ for all } P \in perm(n) \tag{3}$$

In other words, S(G, H) is a n-by-n matrix whose (i,j)-entry expressing some kinds of similarity between j-th node of graph G and i-th node of graph H. (DO NOT mistake the order of i and j here)

Note 1: formula (2) is called a template either than an algorithm, because the function S is left to be open, and each implementation of S will provide corresponding node-similarity, and then matching algorithm.

Note 2: formula (3) states that the similarity of two nodes is independent with the order of the node in the graph.

Note 3: formula (2) can be easily solved by the Hungarian algorithm [8].

So a unified framework to generate and analyze graph matching algorithm is constructed based on the implementation of node-similarity function S.

4 Some Algorithms of NSGM Template

In this section, some algorithms of the NSGM template will be introduced. We use several Matlab command as denotations that supposing T is any symbols expressing a matrix, then T(k,:) denotes the k-th row of T, G(:,k) denotes k-th column, and T(i,j) denotes the ij-th entry of T. $\otimes$ denotes the kronecker product operator and vec() denotes the vectorization operator. sum(), sort(), poly(), and diag() have the same meaning with corresponding Matlab functions.

4.1 Directly Constructing NSGM Algorithms

Theorem 1: The following functions are all node-similarity functions.

$S_1(G,H)(i,j) = -|G(j,j) - H(i,i)|$

$S_2(G,H)(i,j) = \sum_{k=1}^{n}\sum_{l=1}^{n} G(k,j) \times H(l,i)$

$S_3 = \mathrm{vec}((G \otimes H + G^T \otimes H^T)^m \times \mathbf{1}_{n^2 \times 1})$, where m can be any natural number and $\mathbf{1}_{n^2 \times 1}$ is a column vector with all n^2 elements 1.

The proof can be provided by verifying that formula (3) holds for S_1, S_2 and S_3, where 1) and 2) is elementary and 3) can be simply proved by induction on m. Details are omitted here.

Note 1: the corresponding node-similarity matching algorithm of function S_2 is actually equivalent to the least square kronecker product graph matching algorithm of Wyk [4].

Note 2: the corresponding node-similarity matching algorithm of function S_3 is actually equivalent to the hubs and authorities matching algorithm of Kleinberg [10].

Obviously, LSKPGM and HAGM algorithms have been re-interpreted in a simple and unified form.

4.2 Constructing NSGM Algorithms by Node-attribute Functions

From Theorem 1, it can be seen that to construct a node-similarity function, one has to consider a graph pair G and H at the same time, which sometimes

makes the construction a little complicated. In this section a new kind of functions called node-attribute functions will be introduced to simplify the construction.

Definition 1: A node-attribute function is a function $f: R^{n\times n} \to R^{n\times m}$ which satisfies

$$f(PGP^T) = Pf(G), \text{ for all } G \in R^{n\times n} \text{ and } P \in \text{perm}(n) \tag{4}$$

Intuitively, f can be seen as a function mapping the edge attribute of graph G to its node attribute.

We shall give some examples of the node-attribute functions.

Theorem 2 : The following functions are all node-attribute functions:

$f_4(G)(k,:)=[\text{sum}(G(k,:)), \text{sum}(G(:,k))]$
$f_5(G)(k,:)=[\text{sort}(G(k,:)), \text{sort}(G(:,k))]$
$f_6(G)(k,:)=[\text{poly}(G(k,:)), \text{poly}(G(:,k))]$

$$f_7(G)(:,k)=\text{diag}\left(\frac{G^k}{n^k}\right), \quad k=1,2,\ldots,n.$$

The proof can be provided by verifying that formula (4) holds for f_4-f_7, which are elementary and are omitted here.

Theorem 3: Function f_8 satisfies formula (4) if the matrix G' (defined below) only has single eigenvalues.

$f_8(G)=A$, where A is calculated by 3 steps:

8.1) $G' := \frac{G+G^T}{2} + \frac{G-G^T}{2}\sqrt{-1}$

8.2) Calculating the eigen-decomposition of the matrix $G'=UDU^*$, where D is the diagonal matrix of eigenvalues in descending order.

8.3) $A = \text{abs}(U)$, which is the matrix whose entries are the absolute value of corresponding entries of U.

The proof is trivial.

Note: the function f_8 doesn't satisfy formula (4) for general case if the matrix G' has multiple eigenvalues.

Using the node-attribute function, one can easily construct the corresponding node-similarity function by the following theorem.

Theorem 4: Let f be a node-attribute function, and S_f is defined as:

$$S_f(G,H) = f(H) \times f(G)^T$$

Then S_f is a node-similarity function.

Followed by Theorem 4, each node-attribute function in this Theorem defines a corresponding node-similarity function denoted as S_4, to S_8. The matching algorithm by node-similarity functions S_6 and S_8 are exactly the symmetric polynomials transformation graph matching algorithm SPGM of Almohamad [1] and the eigen-decomposition method EDGM of Umeyama [17] respectively.

4.3 Testing of NSGM Algorithms

In this section, these algorithms generated by node-similarity function from S_1 to S_8 are numerically compared. In each test, 500 pairs of isomorphic matrices are generated by Matlab, whose elements are uniformly random numbers in [0, 1]. For each pair G and H, a perturbation matrix E is added to Graph H, where every elements of E is a uniformly random number in $[0, \varepsilon]$.

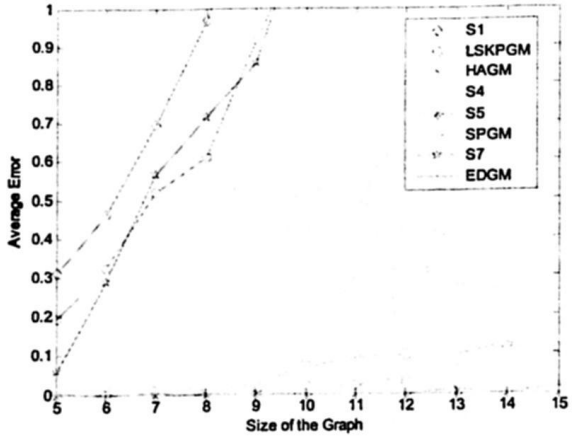


Fig. 1. Average error for ε =0.05.

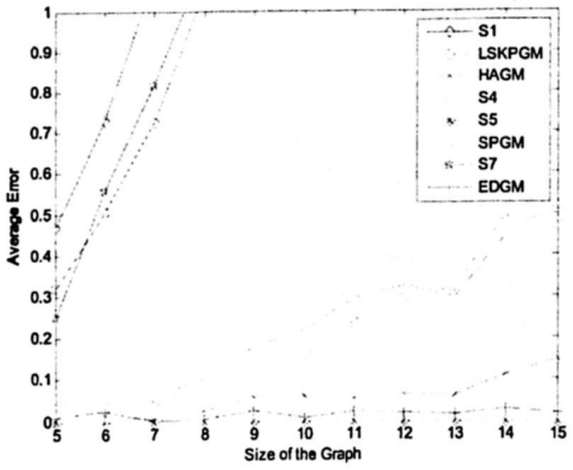


Fig. 2. Average error for ε =0.10

Then the average of matching error for these algorithms are calculated and compared. The matching error is defined as:

$$er(S,G,H) = \left\| P_S G P_S^T - H \right\|_F - \left\| P_0 G P_0^T - H \right\|_F$$

where P_0 is the best matching permutation of G and H, P_S is the matching result calculated by node-similarity algorithm corresponding to node-similarity function S.

The test result for ε =0.05, 0.10 and 0.15 are shown in Fig 1-Fig 3.

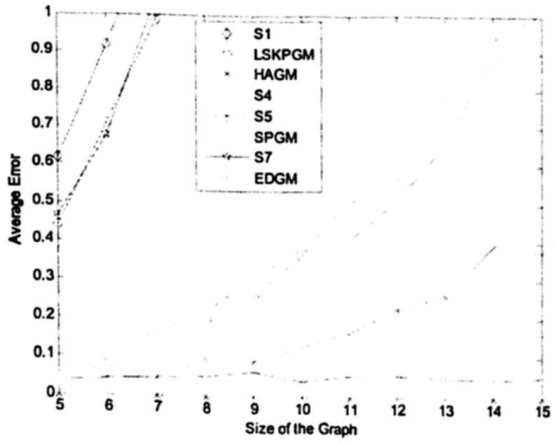


Fig. 3. Average error for ε =0.15

These eight algorithms are clearly in three groups: (1) inaccurate algorithms including S_1, S_7 and LSKPGM; (2) roughly accurate algorithms including S_4 and SPGM; (3) accurate algorithms including: HAGM, EDGM and S_5. The algorithm of S_5 based on simply sorting the matrix elements, gets almost zero-error in all three tests, which in one respect indicates that complex computations usually cover the essence of the problem instead of revealing the essence.

4.4 Computational Complexity

All these algorithms have two main steps, the calculation of node-similarity and the calculation of permutation by Hungarian algorithm. The latter's computational is well known as $O(n^3)$, where n is the number of nodes in graph G and H.

Table 1. Computational complexity of NSGM algorithms

Complexity / Algorithm	Calculating node-similarity	Calculating permutation matrix	In All
S1	$O(n^2)$	$O(n^3)$	$O(n^3)$
LKPGM	$O(n^4)$	$O(n^3)$	$O(n^4)$
HAGM	$O(n^3m)$	$O(n^3)$	$O(n^3m)$
S4	$O(n^3)$	$O(n^3)$	$O(n^3)$
S5	$O(n^3)$	$O(n^3)$	$O(n^3)$
SPGM	$O(n^3)$	$O(n^3)$	$O(n^3)$
S7	$O(n^4)$	$O(n^3)$	$O(n^4)$
EDGM	$O(n^3)$	$O(n^3)$	$O(n^3)$

5 Extension

In the discussion above, only the weighted graphs are considered. But at some circumstance, we have to deal with the multi-weighted graph matching problems which can be defined as:

$$\arg\min_{P\in perm(n)} \sum_{i=1}^{m} \left\| PG_iP^T - H_i \right\|_F$$

where the G_i and H_i are the i-th weight matrices of graph G and H.

The node-similarity algorithm template can be directly expanded for these multi-weighted graphs as:

$$\arg\min_{P\in perm(n)} \left\| P[I,I,\cdots,I] - [S(G_1,H_1),S(G_2,H_2),\cdots,S(G_m,H_m)] \right\|_F$$

where I is the n-by-n identity matrix.

From above, we can see that to solve a multi-weighted graph matching problem using the node-similarity algorithm is as simple as the normal weighted graph matching problem.

6 Limitation of NSGMT

Let S be a node-similarity function satisfying (3)

$S(PGP^T, H) = S(G, H)P^T$, for all $P \in perm(n)$.

Specially, we choose $P_0 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}$

Graph G is also chosen to be special that satisfying

$$P_0 G P_0^T = G \tag{5}$$

Then from formula (3), we get

$$S(G,H) = S(P_0 G P_0^T, H) = S(G,H) P_0^T \tag{6}$$

Which means all columns of S(G, H) are equal

$$S(G,H) = [v, v, \cdots v] \tag{7}$$

Then we get the following theorem:

Theorem 5: Let graph G be a circle, then G satisfies formula (4) and (6). In this case, any permutation is an optimal solution of formula (2). The node-similarity algorithm fails to work.

Theorem 5 only claims that NSGMT are not applicable for circles, how about others? In fact, the essence of this failure is that if graph G is self-similar, which means there is a non-trivial automorphism of graph G, the NDGMT will fail to distinguish those similar nodes of G.

7 Conclusion

In this paper, a unified framework for generating, testing, analyzing, comparing and expanding the node-similarity graph matching algorithms are presented. On one hand, this work provides a new view on those traditional graph matching algorithms based on different theories to see them consistently. On the other hand, this work also point out the essential limitation of solving graph matching problems by simply comparisons of node similarity. This gives us an important enlightment for developing new matching algorithms beyond this algorithm template.

References

1. Almohamad, H.A.: A Polynomial Transform for Matching Pairs of Weighted Graphs. Applied Mathematical Modelling, Vol. 15, pp. 216-222. Elsevier Science (1991)

2. Almohamad, H.A. and Duffuaa, S.O.: A Linear Programming Approach for the Weighted Graph Matching Problem. IEEE Trans. PAMI, Vol. 15, pp. 522-525 (1993)
3. Bai, X., Yu, H. and Hancock, E.R.: Graph matching using spectral embedding and alignment. Pattern Recognition, Vol. 3, Issue 23-26, pp. 398 – 401 (2004)
4. Barend Jacobus van, Wyk.: Kronecker Product Successive Projection and Related Graph Matching Algorithms. Ph.D. diss., University of the Witwatersrand, Johannesburg (2002)
5. Blondel, V., Gajardo, A., Heymans, M., Senellart, P. and Van Dooren, P.: A measure of similarity between graph vertices: applications to Synonym Extraction and Web Searching. SIAM Rev., Vol. 46, No. 4, pp. 647-666 (2004)
6. Finch, A.M., Wilson R.C. and Hancock, E.R.: Matching Delaunay Triangulations by Probabilistic Relaxation. Proc. of Computer Analysis of Images and Patterns, pp. 350-358 (1995)
7. Gold, S. and Rangarajan, A.: A Graduated Assignment Algorithm for Graph Matching. IEEE Trans. PAMI, Vol. 18, pp. 377-388 (1996)
8. Harold, W.K.: The Hungarian Method for the Assignment Problem. Naval Research Logistic Quarterly, Vol. 2, pp. 83-97 (1995)
9. Hummel, R. and Zuker, S.: On the Foundations of Relaxation Labeling Processes. IEEE Trans. PAMI 5, pp. 267-287 (1983)
10. Kleinberg, J.: Authoritative sources in a hyperlinked environment. Journal of the ACM, Vol. 46, No. 5, pp. 604-632 (1999)
11. Kremar, M. and Dhawan, A.P.: Application of Genetic Algorithms in Graph Matching. Proc. of the International Conference on Neural Networks 6, pp. 3872-3876 (1994)
12. Neuhaus, M. amd Bunke, H.: A Quadratic Programming Approach to the Graph Edit Distance Problem. Journal of Lecture Notes in Computer Science, Vol. 4538, pp. 92-102 (2007)
13. Sanfeliu, A. and Fu, K.S.: A Distance Measure between Attributed Relational Graphs for Pattern Recognition. IEEE Trans. SMC 13, pp. 53-363 (1983)
14. Suganthan, P., Teoh, E. and Mital, D.: Pattern Recognition by Graph Matching Using the Potts MFT Neural Networks. Pattern Recognition 28, pp. 997-1009 (1995)
15. Tasi, W.H. and Fu, K.S.: Error-Correcting Isomorphisms of Attributed Relational Graphs for Pattern Recognition. IEEE Trans. SMC 9, pp.757-768 (1979)
16. Ullman, J.R.: An Algorithm for Subgraph Isomorphism. Journal of the Association for Computing Machinery Vol. 23, pp. 31-42 (1976)
17. Umeyama, S.: An Eigendecomposition Approach to Weighted Graph Matching Problems. IEEE Trans. PAMI, Vol. 10, pp. 695-703 (1988)
18. Zhao, G., Luo, B., Tang J. and Ma, J.: Using Eigen-Decomposition Method for Weighted Graph Matching. Lecture Notes in Computer Science Vol. 4681, pp. 1283-1294 (2007)

Thresholding Method based on the Hmax and Hmin Morphological Operators

Edgardo Felipe-Riveron and David Suarez-Hernandez

Center for Computing Research, National Polytechnic Institute,
Juan de Dios Batiz w/n, Col. Nueva Industrial Vallejo, P.O. 07738, Mexico
edgardo@cic.ipn.mx ; davidsuher@yahoo.com.mx

Abstract. This paper presents a new method for thresholding gray level images based on intrinsic properties of the h-extrema Hmax and Hmin morphological operators. Hmax is used to segment the dark objects and Hmin to segment bright objects. Gray levels have a more extended influence over neighbor pixels because all intermediate peaks (troughs) in the original image f are eliminated in a natural way by the process of reconstruction by dilation (erosion) of f from the $f - h$ $(f + h)$ image. It produces a kind of plateau around all pixels that maintain the category of extrema. In practice, our thresholding method includes the pixels located throughout the natural ramp-shaped edge commonly present between adjacent regions having any two different gray levels.

Keywords: . Thresholding, . Hmax . morphological . operator, . Hmin morphological operator, image segmentation.

1 Introduction

Thresholding is a simple and direct method to obtain a binary image from a gray level image. Binary images make easier the description of objects resulting from the thresholding process. In a binary image objects appear black in a white background or vice versa. Thresholding is the simplest procedure for segmenting images. From the programming point of view, it is computationally inexpensive and fast [1] [2].

In principle, any gray level value used as a threshold produces a binary image from a gray level image. However, not every threshold when applied to the gray level image produces a useful binary image. From a gray level image, which can be considered the best thresholding method for obtaining a useful binary image? How can we obtain the best value for the threshold? The answers to these questions are not trivial in any way, because in general the answer depends on the complexity (contents) and on the particular characteristics (contrast, type and rate of the noise, homogeneity of the illumination, and many others) of the original gray level image, and also on the objects we would like stand out in the binary image. As a conclusion, the selection of the appropriate threshold depends on the object(s) we would like

Advances in Artificial Intelligence: Algorithms and Applications
Research in Computing Science 40, 2008, pp. 83-94

to isolate from the gray level image. Nevertheless, sometimes it is impossible isolate successfully all desired object from a gray level image using the thresholding. Generally, the global statistical characteristic between *gray levels* versus *probability of occurrence* of the image, known as the histogram, is used for selecting the threshold value. There is not a general method for selecting the optimal threshold value in order to get always the best or most useful binary image. In general, the determination of the thresholding value in a non-supervised manner could be done automatically when for a given application we can assure that all images will have similar histograms. For the sake of clarity in this paper the threshold is selected in a supervised way.

Every digital image when digitized acquires random pixel values in a very broad range of levels due to the intrinsic noise produced by a deficient illumination or by non-uniform input device parameters. All gray level images used in this paper to explain the method have not been preprocessed in any way; in other words, they are shown just as they were acquired. This means that images probably have a certain amount of additive noise with a Gaussian, uniform, or random distribution. On the other hand, the jump from regions with bright pixels to others with dark pixels or vice versa is never drastic. In the zone between the bright and dark regions in an image they are found always pixels with gray levels forming a ramp-shaped edge. Due to these two problems mentioned above, the selection of the most suitable threshold to get a useful binary image from a gray level one could be a very difficult and cumbersome task.

In our study we consider two main types of binary images of interest for application researchers and developers: the first type show independent objects clearly distinguished in a contrasting background (Section 2); this is the case in images with a bimodal histogram (Fig. 1) and the process executes a complete segmentation. The second one occurs when a given number of the darkest (or brightest) pixels (in general many pixels) must be clearly distinguished from the remainder in the image (Section 3); this is the case in images with a multimodal histogram (Fig. 2) and the process executes a partial segmentation [2]. These two types of binary images are obtained preferably using only one threshold applied to the original image. From the first type, it is possible later to count the objects after labeling and to calculate for each one the area, perimeter, and other geometrical characteristics in order to classify them according to their size, orientation, etc. In the second type of binary image, the interest could be simply the separation of the darkest (or the brightest) pixels (objects) in order to segment them from the remainder to guide a robot throughout a contrasted line in the floor, for detecting holes in a road, or for locating the position of luminous sources illuminating the scene, reflexes, shadows, among others aims.

Figure 2 shows the image of a house and three binary images obtained from the original in three different arbitrarily manually selected thresholds, Th, in gray levels 141, 169 and 226, respectively (indicated with the arrows), where appear the minima in the histogram in Fig. 3.

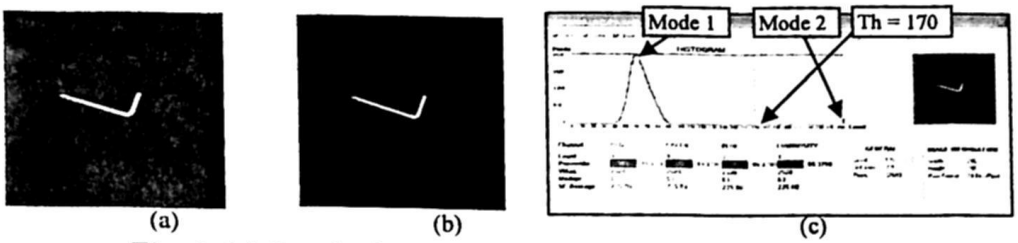


Fig. 1. (a) Gray level image; (b) Binary image; (c) Bimodal histogram

According to the histogram, all values are consecutive relative minima with 16, 7 and 6 pixels, respectively. First two cases clearly segment the darkest zones of the original image, and the third one, segments exclusively the brightest parts of the house. However, how can we assure that these manually selected thresholds are the most adequate? Which from the three thresholds is better to segment the darkest (or brightest) part of the house (or of the whole image)? Which threshold segments the house with a maximum number of dark or bright pixels? How can we select the adequate threshold to segment only the darkest zones of the house (shadows)? Or only the brightest parts of the house? It is practically impossible to find a categorical correct answer to these questions. For the sake of comparing results, from the total 65536 pixels (px), the number of dark and white pixels in three images is shown in Table 1.

Fig. 2. (a) House. Three resulting binary images of the house from three arbitrarily selected threshold levels, Th, in: (b) 141; (c) 169; (d) 226, shown by the arrows in the histogram in Fig. 3

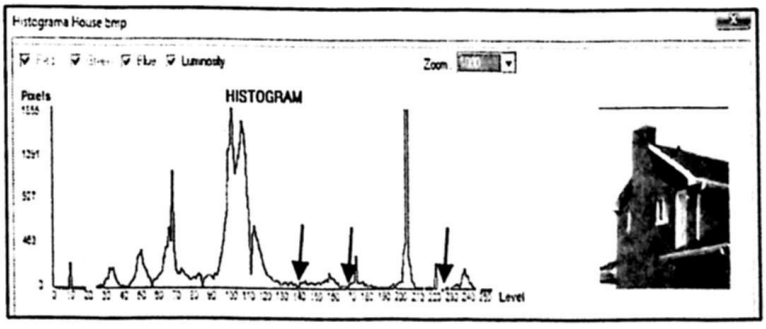


Fig. 3. The histogram of the House

Table 1. Number of white and black pixels in binary images shown in Fig. 2 with the three threshold levels selected arbitrarily

Image	Threshold (Th)	White px	Dark px
b	141	26792	38744
c	169	24498	41038
d	226	1504	64032

2 The Problem

The main purpose of this paper is to explain a relatively simple method to determine the most suitable global threshold that transform a gray level image in a binary one. In this paper a morphological method is proposed to get, in a simple and straightforward way, a more exact global threshold value in gray level images having either a bimodal or a multimodal histogram. The procedure is based on the use of the h-extrema morphological operators, hmax to isolate the dark objects and hmin to isolate the bright objects.

3 Brief State-of-the-Art

Thresholding in image processing is not new. Here, we discuss thresholding methods (pixels-based) requiring only one threshold level and based on the histogram to give as result a binary image. In general, the pixels of the output $g(i, j)$ obtained from the thresholding process fulfils the following conditions [2]:

$$g(i,j)=1 \quad for \quad f(i,j)>T$$
$$g(i,j)=0 \quad for \quad f(i,j)\leq T$$

Where $f(i, j)$ is the original gray level image, $g(i, j)$ is the output binary image and T is the threshold level selected. Brightest pixels are related either to the objects and darkest pixels is related to the background, or vice versa. Many times the threshold is selected by trial and error but it is not obtained the best solution. Frequently, there are many thresholds that produce useful binary images.

There are many classical methods to obtain a binary image using only one threshold level T. Amongst them the following methods are commonly used.

In the variable or adaptive thresholding the image f is divided into subimages f_c [2]. A different threshold is determined independently in each subimage. If the threshold cannot be determined in some subimage, it can be interpolated from thresholds determined in neighboring subimages. Each subimage is then processed with respect to its local threshold.

$$T=T(f,f_c)$$

Optimal thresholding is based on the approximation in the histogram of an image using a weighted sum of two or more probability densities with normal distribution. The threshold is set as the closest gray level corresponding to the minimum probability between the maxima of two or more normal distributions, which results in minimum error segmentation [2], [7].

There are some other methods based on the entropy [3], [4], [5], [8]; on fuzzy sets [6]; on minimum error [7]. There has been also revised the survey [4].

The Otsu method [11] is a commonly used thresholding method. It is a nonparametric and unsupervised method of automatic threshold selection. An optimal threshold is selected by the discriminant criterion of maximizing the separability of the resultant classes in gray levels. The procedure utilizes the zeroth- and the first-order cumulative moments of the gray level histogram. However, the method has failed when the difference between the levels of the objects and the background is small.

Many relatively recent morphological solutions appear in the literature to select the best threshold in gray level images to turn it out a binary one [13]-[19]. However, any of them uses the hmax and hmin operators to carry out thresholding.

4 The H-extrema

The regional extrema of a raw image mark relevant as well as irrelevant image features. H-extrema transformations provide us with a tool to filter the image extrema using a contrast criterion. More precisely, the h-maxima transformation suppresses all maxima whose depth is lower or equal to a given threshold level h. This is achieved by performing the reconstruction by dilation of f, R_f^{δ}, from $f - h$ [12]:

$$h\max = HMAX_h(f) = R_f^{\delta}(f - h) \tag{5}$$

The h-maxima transformation is illustrated in Fig. 4 on a 1-D signal.

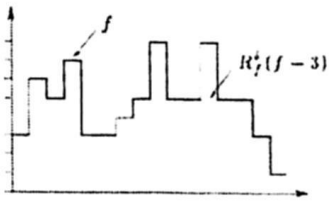


Fig. 4. H-maxima transformation of a 1-D signal using a contrast value of 3 intensity levels

The h-minima transformation is defined by analogy as:

$$h\min = HMIN_h(f) = R_f^{\varepsilon}(f + h) \tag{6}$$

Where R^{*}_{f} is the reconstruction by erosion of f.

The size of the structuring element used in these operations, is the elemental one, that is, the smaller commonly used by the processes of reconstruction. In our case, we used a 3 x 3 square flat structure element.

5 The Method

The method for thresholding gray level images could be supervised or non-supervised. In this paper we explain the supervised version of the method. Also, the particular method varies if the user is interested in selecting the darkest or the brightest objects (or zones). The most straightforward and easy way depends on the number of gray levels of dark (or bright) pixels we can appreciate in the image. The exact division between which gray levels can be considered as dark or bright in a particular image is not so easy, although obviously the best results are obtained when the gray level image has a good contrast.

In our method, if the user wants to threshold the dark objects (or dark zones) in the image, then he/she must use the hmax operator; conversely, if the user wants to threshold the bright objects (or bright zones) in the image, then he/she must use the hmin operator. Commonly, it will be a good practice to consider either the dark or the bright objects, depending on the number of pixels of each class appearing more abundantly in the image.

When we apply the operator hmax to threshold the darkest objects in the image, the histogram of the *modified* image shifts to the left in the value h, according to Eq. (5), because all pixels of the original image is subtracted in the value h. For that reason, in the histogram of the *modified* image, obtained when the operator hmax is applied, a maximum gray level appears always in the value (255 – h). For this reason, maxima over this value in the histogram of the original image are not of importance. On the other hand, when we apply the operator hmin to threshold the brightest objects in the image, the histogram of the *modified* image shifts to the right in the value h, according to Eq. (6), because all pixels of the original image is added the value h. In it a maximum gray level appears always in the value h. For this reason, maxima below h in the histogram of the original image are not of importance.

The general methodology to select the most adequate threshold value for segmenting dark or bright objects from a gray level image using the hmax and hmin operators is the following:

1. From the histogram of the original image select the first relative maximum (from left to right in the histogram) as the value of h. Gray levels with the relative maxima having few pixels (less than 100 pixels in images with thousands of pixels, 0.05%) at the beginning of the histogram are excluded, because they do not contribute appreciably with a significant difference causing that the objects appear very scarce in the

resulting binary images. Also, the common maximum and minimum in levels 0 and 255 neither are used by our method.

2. For dark (bright) objects apply the operator hmax (hmin) to the original image to get a *modified* image.
3. From the histogram of the *modified* image select the eventually appropriate threshold values from the relative minima and apply them to the original image. If hmax was used, the minima to be selected are located nearest to the left part of the histogram of the *modified* image. If hmin was used, the minima to be selected are located nearest to the right part of the histogram of the *modified* image.
4. The results of the previous step may be many binary images, one for each minimum selected from the *modified* image, each with a given number of dark (bright) objects.

We consider heuristically the best threshold value the gray level which produces a binary image with more spatially independent (non-connected) objects together with a greater total amount of dark (or bright) object pixels.

The most problematic step in the methodology given above is the steps 3. If the user applies hmax (hmin) and he/she is going correspondently to threshold dark (bright) objects, then he/she must select the best grey level from all minima in the histogram of the *modified* image. To extract the adequate relative minima from the histogram of the *modified* image to the histogram was explored from left to right with a 41-pixel window. The window size of 41 demonstrated that is the best size in our practice for extracting the minimum

6 Results and Discussion

From the experiments carried out we will compare the resulting binary image obtained by the hmax(hmin) method with those obtained by other methods. Applying the method indicated above to the image in Fig. 1a (thresholding of the first type), the following results were obtained (Fig. 5 to Fig. 9).

Fig. 5 shows the original gray level image and its histogram. From the histogram, we can observe that the first relative (absolute in this case) maximum is located in gray level 59. This is the value of h (step 1). Due to the object to be segmented is bright we will apply to the original image the operator hmin. Then we obtain the *modified* hmin image shown in Fig. 6a (step 2). Figure 6b shows the histogram of the *modified* hmin image and Fig. 6c shows a fragment of the same histogram (zoomed) with an arrow indicating the first minimum, from right to left, at the gray level 99 (step 3).

Considering the gray level 99 as a threshold, when it is applied to the original image we obtain the binary image shown in Fig. 6d (step 4). Figure 8a shows the binary image obtained in the next minimum located in gray level Th = 143. Figure 7a shows the difference between the binary images shown in Fig. 6d and Fig. 7a. Undoubtedly, the quality of the segmented object

resulting in Th = 99 is much better than that obtained in Th = 143. This demonstrates that the proposed method incorporates to the final segmented object some pixels from the diffuse edge between the bright and dark pixels on account of some dark pixels from the background. Similarly, we can asses that the use of the hmin method assure to select the threshold value in a more straightforward way than with a simple inspection.

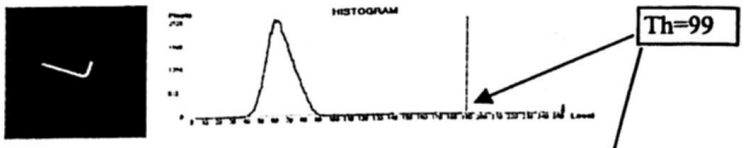


Fig. 5. Original gray level image and its corresponding histogram

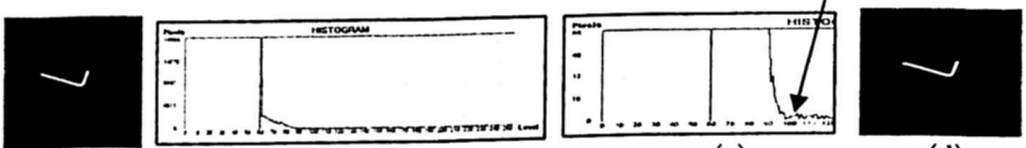


Fig. 6. (a) Hmin image; (b) Histogram of the *modified* hmin image; (c) Histogram zoomed (d) Binary image with Th = 99

Fig. 7. (a) Binary image with threshold in Th = 143; (b) Difference between images in Fig.6d and 7a

Figure 8a shows the resulting binary image when the threshold Th = 99 is applied to the original image; Figure 8b shows the binary image obtained by the Otsu method [11]; Figure 8c shows the 55 pixels in excess obtained by the hmin(hmax) method when it is compared with the result obtained by the Otsu method. This demonstrate that the hmin(hmax) method described in this paper presents better efficiency, from the point of view of the number of total pixels in the minimum segmented objects, than the Otsu method. Taking into account the results obtained (more white pixels for the bright object), the hmin(hmax) method selects in each case a better threshold value in front of all other possible threshold.

Fig. 8. (a) Binary image obtained from the original thresholded in Th = 99; (b) Binary image obtained from the original thresholded by Otsu method; (c) Difference in excess (55 px) between (a) and (b)

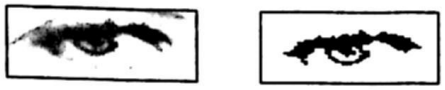


Fig. 9. Detail of the right eye of Lenna (pupil) segmented from the original image

Table 2. Some other results in images with dark objects having trimodal and multimodal histogram

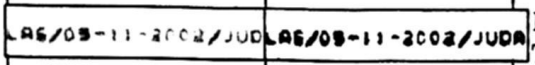
No.	Name	Original image	Binary image	(HMax) Mh; Th	Histogram Type
1	Runner (green plane)			mh = 61 Th = 166	TM
2	Runner (green plane)			Mh = 61 Th = 71	TM
3	Lenna (red plane)			Mh = 96 Th = 125	MM
4	Road			mh = 230 Th = 241	MM
5	Micro-text			MH = 85 Th = 170	BM

Table 2 shows other results obtained after the application of the operator hmax to detect dark or bright objects in gray level images having bimodal (BM), trimodal (TM) and multimodal (MM) histograms. Binary image 1 shows the fingers of the runners at the same time the wide white lines on the floor are shown. Image 2 shows only the shadow of the runner. Figure 9 shows a detail in the right eye of Lenna (the pupil) shown as the image 3, detected

by the proposed method, with Mh = 96 and Th = 125. It is worth to note that the detection was achieved in the complete original image (not in the fragment shown). Image 4 shows the white objects found on the image of the road: the sky, four posts of the fence and the lines on the pavement. Finally, the binary image 5 shows clearly the holes in letters A and D, and in the number 0 (zero) in the image of the micro-text.

7 Advantages of the Method

The method, considered as one of local thresholding, has the following advantages:

1. Gray levels have an extended influence over neighbor pixels because all intermediate peaks (troughs) in the original image f are eliminated in a natural way by the reconstruction by dilation (erosion) of f from the $f - \mathrm{h}$ ($f + \mathrm{h}$) image. It produces a kind of plateau around all pixels that maintain the category of extrema.
2. In practice, thresholding by our method includes some of the pixels located throughout the natural ramp-shaped edge commonly created between two different gray levels, making possible to add more pixels to dark (bright) objects appearing in the original image.
3. Additive (either uniform or Gaussian) noise, intrinsically and frequently present in digital images, has less presence in the so called *modified* image. This produces more homogeneous gray level distribution, either related to objects or to the background, when the minima for thresholding are selected. For this reason, the quality of the final binary image is highly improved, that is, isolated objects with some few pixels do not appear in the final image as artifacts.
4. Because of the discrete nature in the selection of the threshold gray level, the thresholding has effect each time over a broader range of gray levels than with any other method used for thresholding.
5. Details of interest in the original images to be thresholded have less probability to be lost in the resulting binary image (See the pupil of Lenna in Fig. 9 and the holes of the micro-text image in Table 2). This is accomplished using as the threshold in the original image the gray level located as the first minimum in the *modified* hmax image located after the first maximum (from left to right in the histogram) used as h in the hmax operator and as the first minimum in the *modified* hmin image located before the first maximum ((from right to left in the histogram) used as h in the hmin operator.
6. It is achieved a higher selectivity (or discrimination) of the bright (dark) objects, since the thresholding is selected very easily (from less number of minima) on the modified images hmin (hmax), respectively (See the binary images 1 –runner- and the image 4 –road- in Table 2).

7. The quality of the original image (i. e. blurred) is not of great importance because it does not limit in any form to achieve good objects segmentation. It is possible to find in a relatively easy way the most suitable threshold value.

8 Conclusions

The new method proposed for thresholding to convert gray level images in binary images is based on intrinsic properties of the h-extrema hmax and hmin morphological operators. hmax is used to segment the dark objects and hmin to segment bright objects. Gray levels have a more extended influence over neighbor pixels because all intermediate peaks (troughs) in the original image f are eliminated in a natural way by the process of reconstruction by dilation (erosion) of f from the $f - \mathrm{h}$ ($f + \mathrm{h}$) image. In practice, the proposed thresholding method includes many pixels located throughout the natural ramp-shaped edge commonly present between adjacent regions having any two different gray levels. It proved also to be more immune to noise.

Acknowledgement. The authors of this paper wish to thank the Computing Research Center (CIC), Mexico; General Coordination of Postgraduate Studies and Research (CGPI), Mexico, and National Polytechnic Institute (IPN), Mexico, for their support.

References

1. Gonzalez R. C. and Woods R. E.: *Digital Image Processing*, 3rd Ed., Prentice Hall, Inc (2008)
2. Sonka M., Hlavac V., Boyle R.: Image Processing, Analysis, and Machine Vision, 2nd. Ed. Brooks and Cole Publishing (1998)
3. Pun, T.: Entropic Thresholding: A new Approach, Computer Vision Graphics Image Processing, 16 (1981) 210-239
4. Sahoo, P. K., Soltani S.,Wang A. K. C.: A survey on thresholding techniques, Computer Vision Graphics Image Processing, 41 (1988) 233-260
5. Kapur, J., Sahoo, P., Wong, A.:.A new method for graylevel picture thresholding using the entropy of the histogram. Comput. Vision Graphics Image Process. 29 (3) (1985) 273–285
6. Yager, R.:.On the measure of fuzziness and negation. Part I: Membership in the unit interval. Int. J. General Systems 5 (1979) 221–229
7. Kittler, J., Illingworth, J.:.Minimum error thresholding. Pattern Recognition 19 (1) (1986) 41–47
8. Abutaleb, A.S.:.Automatic thresholding of gray-level picture using two-dimensional entropy. Comput. Vision, Graphic, Image Process 47 (1989) 22–32
9. Rosin P. L., Ioannidis E.: Evaluation of global image thresholding for change detection. Pattern Recognition Letters 24 (2003) 2345–2356

10. Parker J. R.: Algorithms for Image Processing and Computer Vision, USA, John Wiley & Sons, (1997)
11. Otsu N.: A Threshold Selection Method from Gray level Histograms, IEEE Trans. on System, Man and Cybernetics, Vol. SMC 9, No. 1 (1979) pp. 62-66
12. Soille P.: Morphological Image Analysis, Principles and Applications, 2nd. Ed., Springer (2002)
13. P. A. Kelly, G. Chen Iterative segmentation algorithms using morphological operations, IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP-93, Vol. 5 (1993) pp. 49-52
14. Di Ruberto C., Dempster A., Khan S. et al., Analysis of infected blood cell images using morphological operators, Image and Vision Computing, Vol. 20, Issue 2 (2002) pp.133-146
15. Pujol F. A., Suau P., Pujol M. et al., Selection o fan automated morphological gradient threshold for image segmentation. Application to vision-based path planning. Advances in Artificial Intelligence-Iberamia 2004, Vol. 3315 (2004), pp. 667-676
16. Pen Qiming, Jia Yunde, A Fast Morphological Algorithm for Color Image Multi-Scale Segmentation using Vertex-Collapse, Third International Conference on Image and Graphics (ICIG'04) (2004) pp. 60-63
17. Lopez RAP, Chamizo JMC, Lopez MP, et al. Selection of an automated morphological gradient threshold for image segmentation, Progress in Pattern Recognition, Image Analysis and Applications, Vol. 3287 (2004) pp. 92-99
18. Arnaldo Camara Lara, Robeto Jr. Hirata, Motion Segmentation using Mathematical Morphology, XIX Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI'06) (2006) pp. 315-322
19. Ning Lu, Xizheng Ke, Segmentation Method based on Gray-Scale Morphological Filter and Watershed Algorithm for Touching Objects Image, Fourth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2007), Vol. 3 (2007) pp. 474-478

3-D Fractal Characterization of Tumors from a Computer Tomography Scan

Ernesto Cortés Pérez [1], Tomás Morales Acoltzi [2],
Francisco Viveros Jiménez [3]

[1] Departamento de Ingeniería en Computación, Universidad Del ISTMO, UNISTMO, Campus Tehuantepec, Santo Domingo Tehuantepec, Oaxaca, Mexico
neto_144@bianni.unistmo.edu.mx
[2] Centro de Ciencias de la Atmósfera, UNAM, Circuito Exterior, CU, México DF CP 04510.
acoltzi@atmosfera.unam.mx
[3] Departamento de Licenciatura en Informática, Universidad Del ISTMO, UNISTMO, Campus Ixtepec, Avenida Universidad S/N, Ciudad Ixtepec Oaxaca, CP 70110, Mexico
fviveros@bianni.unistmo.edu.mx

Abstract. The use of the Box-Counting method (BC) to calculate the fractal dimension of invasive pathologies in the human body is proposed in this paper. BC was applied to calculate the fractal dimension of a tumor from medical images of human brains with cancer. The BC test required additional image processing algorithms and 3-D reconstruction software for the processing of a sample area. The BC results were used to determine the size and volume of a tumor; this allows an oncologist to perform a more informed diagnostic.

Keywords: Fractal dimension, Box-Counting, cancer, medicine, tumor size, and pathology size.

1 Introduction

Saving human lives is the primary goal of medicine. Cancer is an invasive disease that is responsible for many deaths annually worldwide. For defeating cancer, an oncologist needs to know the following tumor characteristics:

- Specific type.
- Dimensions.
- Location.
- Internal organs affected.
- Cancer stage.
- Appearance.
- Growth rate.

Advances in Artificial Intelligence: Algorithms and Applications
Research in Computing Science 40, 2008, pp. 95-106

Knowledge of these features allows an oncologist to estimate the operability of the tumor and prognosis of the patient.

Medical technology has improved during the last century. Technology like X-rays, ultrasound, magnetic resonance imaging (MRI) and computer tomography (CT) allows a physician to obtain an internal image of the body. This technology can even show brain functioning and structure. However, it is still difficult for a doctor to obtain an accurate measurement of an invasive pathology without an invasive surgical procedure. For this reason, many investigators have been using computer programs to help doctors to measure and identify more accurately an invasive pathology [1] [2]. This paper shows how Box-Counting method, which is widely used for calculation of fractal dimension, can be used to measure the size of a tumor in a human brain.

Box-Counting needs a digital filtered image for its performance. The process for obtaining this input is explained briefly later in this paper. Figure 1 shows an example of a CT scan of a patient with a brain tumor.

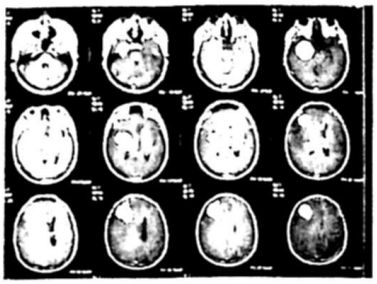


Fig. 1. CT scan of a brain with tumor.

2 Box–Counting Method

This section explains the process for calculating the dimension of an object with Box-Counting. First, the basic geometrical concepts are explained. Next, Box-Counting is explained. Finally, the measurement of teh size of an object with box-counting is shown by using exponential laws.

2.1 Traditional and Fractal Geometry Concepts

Traditional geometry is the sub-discipline of mathematics that studies the features and measurement of elements like points, lines, curves, planes, figures and volumes. However, traditional geometry can not represent the shapes found in nature like mountains, animals, clouds, leaves, trees, etc. Fractal geometry provides a mathematical model for these complicated natural forms (also called abstract forms).

In geometry, the dimension of a space is defined as the minimum number of coordinates needed to define every point within it. All forms have dimensions: points have none, lines have one dimension, surfaces have two dimensions and volumes have three dimensions. For example: two dimensions are required to represent a rectangle, a cube requires three dimensions, etc. This concept of dimension is called topological dimension.

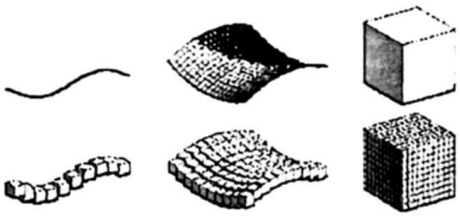


Fig. 2 Geometric shapes: line, surface and cube.

Surfaces like a lung, a brain or a tumor have three dimensions therefore the unevenness of an object can be considered as increment in a specific dimension. Rough curves have between one and two dimensions. Rough surfaces have between two and three dimensions. This concept is called fractal dimension. Fractal dimension is a precise parameter with which we measure the conceptual and visual complexity of an object. Using fractal dimension, the volume of an irregular object can be calculated.

2.2 Box-Counting Insight Explanation

Box-Counting is a method used to obtain the fractal dimension of an object. The Box-Counting steps are:

1. Draw a rectangular grid over the image that contains the object to identify. Each box will have a width and height of r. The grid will have N boxes.
2. Count the boxes that contain the object.

The relation between N and r is given in eq. 1.

$$N(1)=1$$

$$N\left(\tfrac{1}{2}\right)=4=\frac{1}{\left(\frac{1}{4}\right)}=\left(\frac{1}{\left(\frac{1}{2}\right)}\right)^2$$

$$N\left(\tfrac{1}{4}\right)=16=\frac{1}{\left(\frac{1}{16}\right)}=\left(\frac{1}{\left(\frac{1}{4}\right)}\right)^2$$

$$N\left(\tfrac{1}{8}\right)=64=\frac{1}{\left(\frac{1}{64}\right)}=\left(\frac{1}{\left(\frac{1}{8}\right)}\right)^2$$

$$N(r)=\left(\frac{1}{r}\right)^2 \tag{1}$$

The precision of the technique depends directly on the size and number of boxes. A large number of smaller boxes will give a more accurate result. However, more boxes imply more image processing.

2.3 Exponential Laws

Equation 2 represents the relation between N and r for a d dimensional figure.

$$N(r)=\left(\frac{1}{r}\right)^d \tag{2}$$

Equation 3 is obtained by using exponential laws in eq. 2.

$$N(r)=k\left(\frac{1}{r}\right)^d \tag{3}$$

Equation 4 is obtained by applying logarithm laws to eq. 3.

$$\log(N(r))=\log(k)+\log\left(\left(\tfrac{1}{r}\right)^d\right)=d\log\left(\tfrac{1}{r}\right)+\log(k) \tag{4}$$

Equation 5 is obtained applying the limit of r when $\lim_{r\to 0}$. This deduction is made with an expectation of a better estimation.

$$d=\lim_{r\to 0}\frac{\log(N(r))}{\log\left(\frac{1}{r}\right)} \tag{5}$$

If the limit exists in eq. 5, then d can be calculated. However, the limit operation is complex. Equation 6 shows a valid approximation.

$$\log(N(r))=d\log\left(\frac{1}{r}\right)+\log(k) \tag{6}$$

Equation 7 is obtained by replacing the symbols of the linear equation in slope intersect form with eq. 6.

$$y=mx+b \tag{7}$$

where m represents the dimension.

In equation 7, b is the point, where the line intersects with y. A plot of $\log(N(r))$ against $\log(\frac{1}{r})$ must be approximated to the line with slope m. This approximation is called log-log approximation and is commonly used to find the dimension by Box-Counting method.

3 3-D Digital Reconstruction of a Brain Tumor from CT Scans

3-D digital reconstruction of a patient's head and a brain tumor is required before using the Box-Counting Method. This reconstruction gives a good visual representation of the size and position of the pathology.

Three processes are necessary to perform a 3-D digital reconstruction:

1. Segmentation [3] of the desired object from CT scans with Sobel operator [12].
2. Reconstruction of the 3-D model with Delaunay's triangulation [9].
3. Rendering of the resulting model.

3.1 Reconstruction of the Patient's Head

The first step for reconstructing the head is identification of the head's border in the sample images. 20 images form the sample in this research. Figure 3 shows an example of segmentation of a sample after using binary filtering and border detection with Sobel operator [12].

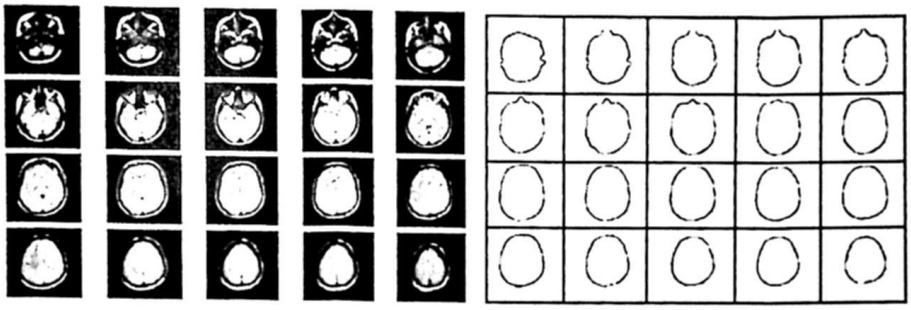


Fig. 3 Segmentation of the sample with Sobel operator.

Once the border is identified in the sample, the images will be superimposed in a 3-D space, as shown in Fig. 4.

Fig. 4 Borders from sample are superimposed in a 3-D space.

Next, Delaunay triangulation is used to create a non-structured grid. This process maximizes the interior angles with great precision (minimal rounding errors). Figure 5 shows the result of Delaunay triangulation in the borders obtained from the sample.

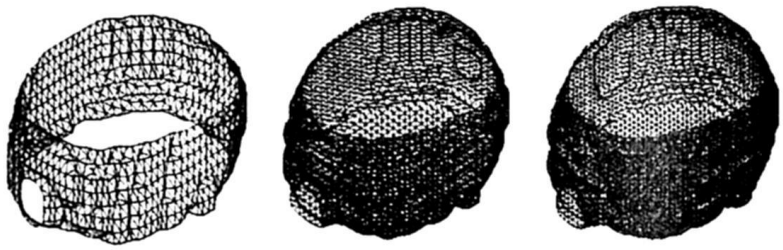


Fig. 5 Results from Delaunay triangulation process.

A rendering process [2] is executed over the grid obtained with Delaunay triangulation. Rendering is a process that creates an image taking environmental effects into account. Figure 6 shows the results of the rendering process.

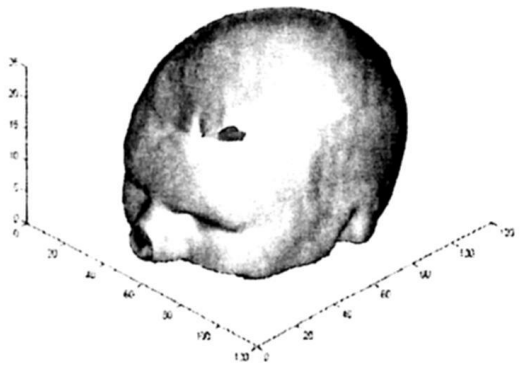


Fig. 6 Results of the rendering process of the patient's head.

3.2. Reconstruction of the Brain Tumor

The process of reconstruction of the brain tumor is exactly the same as the one used on the reconstruction of the patient's head. The first step is applying the segmentation process [3] to the sample. Figures 7, 8 and 9 show the results of the segmentation process.

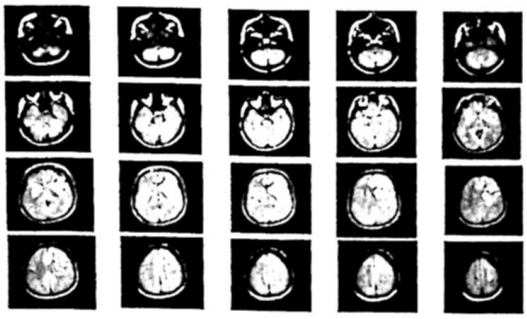


Fig. 7 Segmentation process.

Fig. 8 Segmentation of the pathology under examination

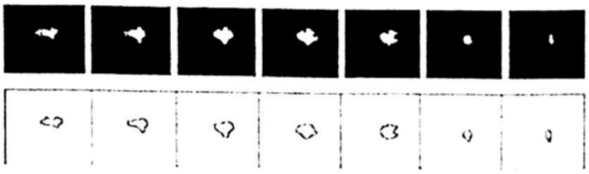


Fig. 9 Tumor borders obtained with Sobel operator.

Finally, the reconstruction of the tumor is applied. The rendering process must be executed taking the patient's head into consideration. In this way, useful information like location, size and internal organs affected is given to the specialist. Figure 10 shows an example. The rendered image can be used by specialists to decide if an operation is viable.

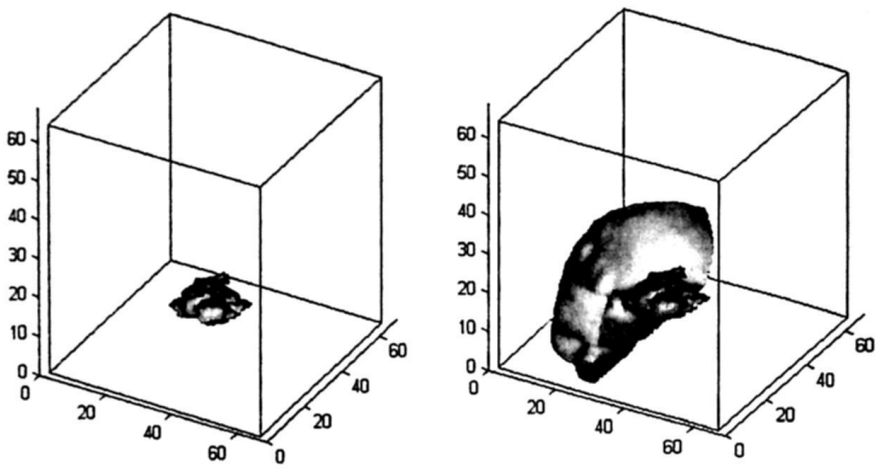


Fig. 10 3-D representation of the tumor.

4 Calculus of the Dimension of the Pathology in a 3-D Space

Measuring the brain tumor with Box-Counting becomes possible after the 3-D reconstruction. Because the object is 3 dimensional, the boxes are going to be cubes and $d = 3$, then $N(r)=\left(\frac{1}{r}\right)^3$.

Box-Counting is an iterative process. In each iteration, the method counts the boxes that contain the pathology. The first iteration has an r value of 1, because $\lim_{r\to 0}$, r is decremented in each cycle. Table 1 show the values obtained in 4 sample iterations of the Box-Counting method. In this table, n is the real depth, width and height of the boxes. Table 1 values consider a 3-D space of *64 x 64 x 64*.

Table 1. Sample iterations of the Box-Counting method

Iteration	n	r	$N(r)$
1	64	1	1/(1/1) ^ 3 = 1
2	32	1/ 2 = 0.5	1/(1/2) ^ 3 = 8
3	16	1/4 = 0.25	1/(1/4) ^ 3 = 64
4	8	1/8=0.125	1/(1/8) ^ 3 = 512

Figure 11 shows a 3-D image of the boxes proposed in table 1. Each iteration has more boxes and therefore more precision.

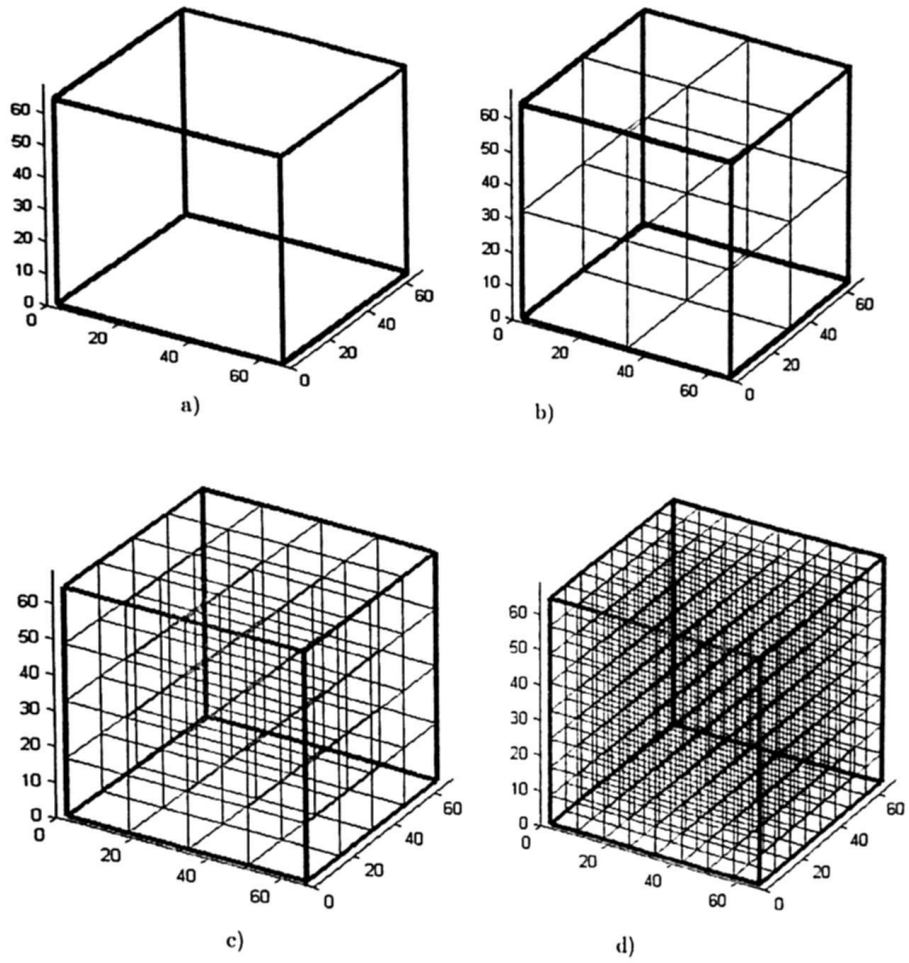


Fig. 11 a) $r_0 = 1$, $N\left(\frac{1}{r_0}\right)^3 = 1$, b) $r_1 = \left(\frac{1}{2}\right)$, $N\left(\frac{1}{r_1}\right)^3 = 8$, c) $r_2 = \left(\frac{1}{4}\right)$, $N\left(\frac{1}{r_2}\right)^3 = 64$, d) $r_3 = \left(\frac{1}{8}\right)$, $N\left(\frac{1}{r_3}\right)^3 = 512$

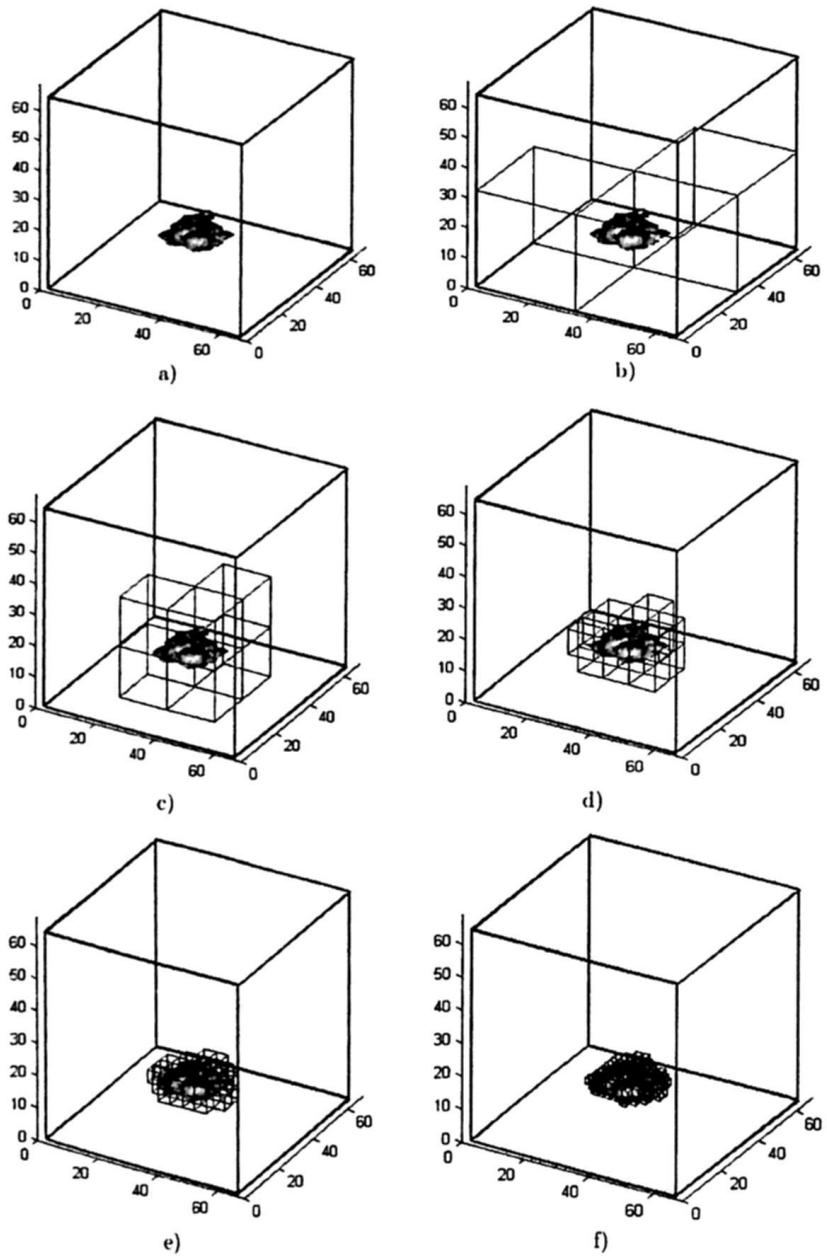


Fig. 12 Stages of the measurement of the tumor

In this research, six iterations were used to measure the brain tumor. Table 2 shows the results of the six test iterations. The results confirm that more precision is gained with smaller values of r.

Table 2. Iterations for measuring the tumor

Iteration	n	r	$N(r)$	Boxes that contain the tumor
1	64	1	1/(1/1) ^ 3 = 1	1
2	32	1/ 2 = 0.5	1/(1/2) ^ 3 = 8	3
3	16	1 / 4 = 0.25	1/(1/4) ^ 3 = 64	6
4	8	1 / 8 = 0.125	1/(1/8) ^ 3 = 512	15
5	4	1 / 16 = 0.0625	1/(1/16) ^ 3 = 4096	42
6	2	1/ 32 = 0.03125	1/(1/32) ^ 3 = 32768	183

The fractal dimension is obtained by using equation 7 and the results are presented in Table 3.

Table 3 Fractal dimension of the tumor.

m = *fractal dimension*	2.4378
B	5.9124

5 Conclusions

The measurement of a brain tumor with Box-Counting in a 3-D space is proposed in this paper. The Box-Counting method requires a prior segmentation process [1, 2, 3] and reconstruction process [7, 8]. The Box-Counting method does not depend on scale of the image as some other methods do. Box-Counting's precision depends on the iteration number. A large iteration number will imply smaller r value, smaller box size, better precision and greater computer resources required.

The fractal dimension obtained (d=2.4378 in test example) provides important information about the position and size of the tumor. This information helps the oncologist to make objective decisions about diagnostics and treatment of the condition.

References

1. Cortés P. Ernesto, "Detección de Cáncer Cerebral: Mediante Computación Inteligente", Memorias del Congreso Internacional de Informática y Computación, ANIEI 2006, XIX, Tuxtla Gutiérrez, Chiapas, Octubre 23-25, ISBN 970-31-0751-6.
2. Cortés P. Ernesto, "Metodología para la extracción de características en imágenes medicas: Tomografías Computarizadas y Resonancias Magnéticas Cerebrales", Memorias del Congreso MICAI 2006 Fifth Mexican Internacional Conference on

Artificial Intelligence, en el Workshop de Tecnologías Inteligentes, Apizaco Tlaxcala, México, Noviembre 13-17, 2006, ISBN 970-94214-1-7.

3. El-Gohary, Awad "Chaos and optimal control of equilibrium states of tumor system with drug" Chaos, Solitons & Fractals (2008), doi:10.1016/j.chaos.2008.02.003
4. El-Gohary, Awad "Chaos and optimal control of cancer self-remission and tumor system steady states" Chaos, Solitons & Fractals, 2008;37(5):1305-16
5. Falconer, K. J. "Techniques in Fractal Geometry" John Wiley & Sons Ltd, 1997
6. Falconer, K. J. "Fractal Geometry-Mathematical Foundations and applications" John Wiley & Sons Ltd, Chichester 1990
7. Canny, John, "A Computational Approach to Edge Detection," IEEE Transactions on Pattern Analysis and Machine Intelligence,Vol. PAMI-8, No. 6, 1986, pp. 679-698.
8. Barber, C. B., D.P. Dobkin, and H.T. Huhdanpaa, "The Quickhull Algorithm for Convex Hulls," ACM Transactions on Mathematical Software, Vol. 22, No. 4, Dec. 1996, p. 469-483.
9. National Science and Technology Research Center for Computation and Visualization of Geometric Structures (The Geometry Center), University of Minnesota, 1993.
10. Lim, Jae S., Two-Dimensional Signal and Image Processing, Englewood Cliffs, NJ, Prentice Hall, 1990, pp. 478-488.
11. Parker, James R., Algorithms for Image Processing and Computer Vision, New York, John Wiley & Sons, Inc., 1997, pp. 23-29.
12. Otsu, N., "A Threshold Selection Method from Gray-Level Histograms," IEEE Transactions on Systems, Man, and Cybernetics, Vol. 9, No. 1, 1979, pp. 62-66.
13. Construcción de mallas Delaunay en tres dimensiones, Enzo A. Dari, Centro Atómico Bariloche, Comisión Nacional de Energía Atómica, Instituto Balseiro, Universidad Nacional de Cuyo y C.N.E.A Consejo Nacional de Investigaciones Científicas y Técnicas.
14. "Triangulacion Delaunay", *Danie Martos López, Ricardo Navarro Moral,* Universidad de Jaén, Departamento de Informática y Geometría Computacional.
15. Zadeh, L. Fuzzy logic, IEEE Computer, 1:83, 1988.
16. Clasificación de Llanto del Bebé Utilizando una Red Neural de Gradiente Conjugado Escalado, J. Orozco García, Carlos A. Reyes García, Instituto Nacional de Astrofísica Óptica y Electrónica.
17. VISIÓN POR COMPUTADOR, Imágenes digitales y aplicaciones, Gonzalo Pajares, Jesús M. de la Cruz, ed. RA-MA.
18. González, R., Woods, R. Digital Image Processing. Adison – Wesley.
19. Li, W. "An equivalent definition of packing dimension and its application" Nonlinear Analysis: Real World Applications. Doi:10.1016/j.nonrwa.2008.02.004

Computer Security

(with Carlos Mex-Perera and Raúl Monroy)

Self-healing and Self-protecting Computing Systems: In the Search of Autonomic Computing

Luis M. Fernández-Carrasco, Hugo Terashima-Marín, Manuel Valenzuela-Rendón

Center for Intelligent Systems
Instituto Tecnológico y de Estudios Superiores de Monterrey
Monterrey, México
{A00789695, terashima, valenzuela}@itesm.mx

Abstract. This document presents a work-in-progress agent-based architecture design to tackle two important autonomic computing features: self-healing and self-protecting. Research in the autonomic computing systems area has seen a great growth in recent years as more complex systems have been developed and more communication among those is needed. However, there is still the lack of an architecture that facilitates the emergence of autonomic computing features, and more specifically, self-protection andself-healing. Consequently, this document aim satfilling this need and proposes an agent-based approach in order to achieve these autonomic computing properties. In order to evaluate the design, a simulation is used. Based on this simulation, it is believed that the proposed approach is a good first step to wards an autonomi ccomputing architecture.

1 Introduction

Richard Murch [1] defines autonomic computing as the ability to manage one's computing enterprise systems through hardware and software so that they automatically and dynamically respond to the requirements of the changing environment. In other words, this means achieving *self-healing*, *self-configuring*, *self-optimizing*, and *self-protecting* hardware and software that behave according to defined policies [1]. These systems are thought after the autonomic nervous system which responds to the needs of the body, so autonomic computing systems should respond to the needs of the environment.

Autonomic computing was born as a consequence of the great advances in networking, computing technologies, and software tools. These sophisticated applications and services are complex, heterogeneous, and dynamic [2]. Moreover, the underlying information infrastructure (e.g., the Internet) globally aggregates large numbers of independent computing and communication resources, data stores, and sensor networks, and is itself complex [2]. This combined scale of complexity, heterogeneity, and dynamism of networks, systems, and applications have made computational and information infrastructures be brittle, unmanageable, and insecure. This situation has made researchers look for a new paradigm

© G. Sidorov (Ed.)
Advances in Artificial Intelligence: Algorithms and Applications
Research in Computing Science 40, 2008, pp. 109-118

for systems and application design. This new vision has been referred to as autonomic computing and is based on strategies used by biological systems when dealing with similar challenges.

Nevertheless, meeting the grand challenges of autonomic computing requires scientific and technological advances in many fields and from many technologies [2]. The goal of this document, therefore, is to provide a new mechanism to achieve autonomic computing systems. This paper proposes an architecture, following a multi-agent approach, capable to provide and support self-healing and self-protection properties. For evaluation purposes, a simulated environment emulating a computer was built.

The following paragraphs present a thorough description of the related work, pointing out differences and similarities. Then, the main idea behind the proposed model is explained so that, when the next section introduces the architecture model, no doubts arise. Right after the model is presented, the simulation used to test the architecture is explained. After, the preliminary results of this work-in-progress research are presented, followed by the the conclusions that can be drawn from the experiments and by some final words about the proposed approach towards autonomic computing.

2 Related Work

A multi-agent approach to autonomic computing is not something new. Jennings [3] has already mentioned the advantages of decomposing problems in terms of decentralized, autonomous agents. Agents add flexibility and high-level of interactions to the design of a system. Thus, it is no surprise to use autonomous agents when trying to build autonomic systems as todays large-scale computing systems get highly distributed with increasingly complex connectivity and interactions.

As a result, there are works that make use of autonomous agents to achieve autonomic systems. Tesauro et al [4] developed *Unity*, a decentralized architecture to enable autonomic computing based on multiple interacting agents called *autonomic elements*. However, Unity covers only some of the wanted self-$\star$ desired properties, namely, goal-driven self-assembly and real-time self-optimization. The proposed architecture in this document aims at supporting not only the ones achieved by Tesauro et al [4], but most self-$\star$ properties an autonomic system should show, making emphasis on *self-optimizing*, and *self-protecting*.

Bonino et al [5] proposed an agent-based autonomic semantic platform. They makes use of the autonomic computing vision and propose *DOSE* [5] to automatically index new resources in response to search failures and auto-detection of low covered conceptual areas when performing tasks on a semantic platform. Although, this work makes use of an agent-based autonomic platform, it is different from the work proposed in this article as it attempts to go beyond using autonomic computing features on a specific area and, on the contrary, aims at being the core that enables autonomic computing features.

Other related works in the autonomic computing field are *OceanStore* [6], which is a global, consistent, highly available persistent data storage system that supports self-healing, self-optimization, self-configuration, self-protection. Also, *Storage Tank* [6] is a multi-platform, universally accessible storage management system. It supports self-optimization, self-healing. *Oceano* [6] facilitates cost effective scalable management of computing resources for software farms. Nevertheless, all the last mentioned works do not follow an agent-based approach. This article's objective is to provide the first results of a work-in-progress software architecture that is capable of not only enabling autonomic behavior.

3 Main Idea for the Architecture

The main idea is to treat everything as an autonomous agent, even the computer. This follows the idea to use autonomic elements suggested by Kephart and Chess [7]. They point out that autonomic elements should function at many levels, from individual computing components such as disk drives to small-scale computing systems such as workstations. Moreover, many ideas developed in the multi-agent systems community, such as automatic group formation, emergent behavior, multi-agent adaptation, and agent coordination could likely be adapted for autonomic computing.

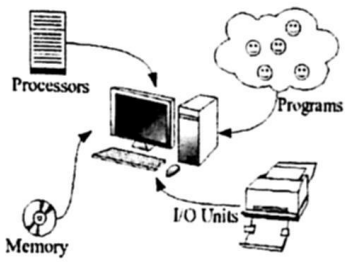


Fig. 1. Agentifying all components.

Consequently, the computer processor, the memory, and even the programs that are run by the computer are modeled as agents, in other words, all components are ***agentified***[1]. These agents communicate with the computer by sending messages as seen in Figure 1. All computing elements have to register with the computer so that this last one *knows* what it has and what it can do with them. The messages that are sent to the computer contain information such as properties, tasks that the component can do, etc.

[1] Agentify is an expression used to turn software and hardware components into agent entities.

4 Proposed Autonomic System Model

This section details the autonomic system model and its components, the relations among them and how they communicate with each other. Graphically, this model is shown in Figure 2. At first glance, one can see that the *Computer* is in charge of managing all other components or modules the autonomic system has available. Therefore, the other components, represented as agents, only know what they are capable of doing, their requirements to work and the results they generate.

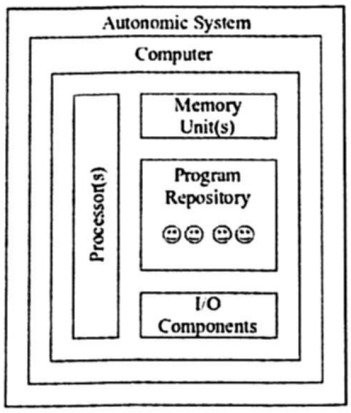


Fig. 2. Autonomic system agent architecture.

Figure 3 presents the communication flow proposed among the principal agents as one program is executed by the autonomic system. Such figure presents all the links among agents and who asks for what at program execution time. The agents that are in Figure 3 are explained in detail lines below.

4.1 Computer Agent

This is the one in charge of organizing all the tasks in the autonomic system. Every new component has to register with the computer as it is *plugged in*. These components send a message, called *Discovery Message*, stating what they can do, what *methods* they have embedded and what output they provide (i.e., a manifest). Once a new component registers with the computer agent, this last one maps into memory the address of the module, its embedded methods and the capabilities that the modules has. Only the computer agent knows where these are and these sections of memory cannot be erased unless the module is *unplugged*.

At start-up, this agent also checks if it has registered all available components to work (i.e., a memory and a processor unit). Once it *knows* it has the available

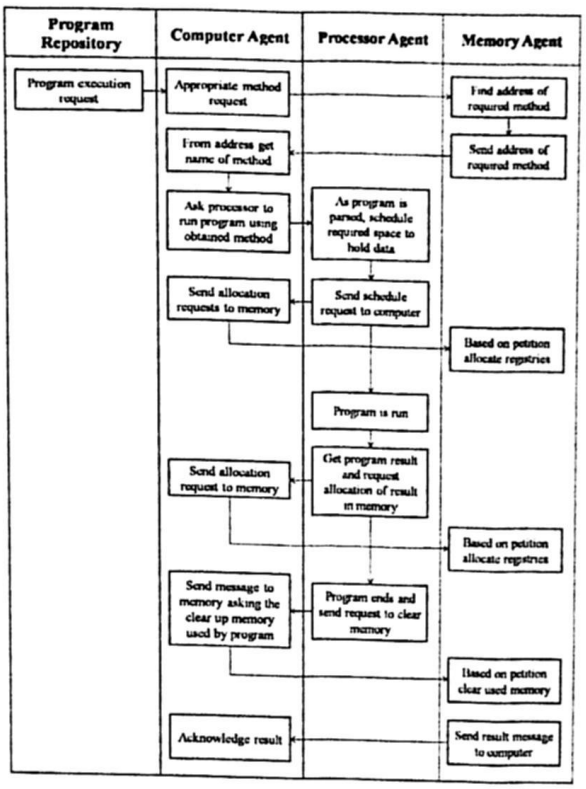


Fig. 3. Communication flow among agents.

resources to respond to petitions, it is able to perform the tasks that other modules send. For instance, if there is the need to run a program, it checks for the appropriate method to conduct that task within the registered processors the computer has. The computer does this by looking at the restricted memory section it has access to. If a suitable method is found, then, it asks the processor to run the program with the method the computer finds suitable. The processor, in turn, asks the computer for memory space. As it was said, only the computer knows what other modules there are in the system and what they are capable of doing. As soon enough memory is allocated for the program to be executed, the processor runs the program and sends back the result to the computer which, then, asks the memory agent to register such result.

4.2 Memory Unit Agents

These agents represent all kinds of storage media. Some examples are hard drives, USB memories, external disks, etc. They can read and write from and to a memory addresses. They can also allocate space for a single value or for an array.

Finally, it is able to erase its contents. However, all the mentioned methods are performed on a request basis, in other words, when the computer agent sends a message asking the memory to perform one action.

4.3 Processor Agents

Agents belonging to this category are the ones in charge to run and perform the tasks the programs demand. In other words, it interprets the programs. It also sends requests for memory access and manipulation to the computer agent. Consequently, it requests to schedule registry readings and writings, or address manipulation. Any processor that is added to the autonomic unit has to register with the computer agent by sending it its manifest. This manifest tells the computer that a new processor has been added and that new capabilities are in the system.

4.4 Program Repository Agent

This agent is the one in charge to spawn programs. A set of programs is already placed in the repository and they are instantiated using a probabilistic method. Let us explain this further, computer users have a set of programs already installed on their PCs; however, there is a number of programs that are used most frequently. Hence, those programs which are used more frequently are instantiated more often. This probability-based instantiation approach is conducted using the genetic algorithm roulette wheel selection method [8]. In other words, this agent acts as the user who is constantly requesting the execution of programs in real-life systems.

4.5 I/O Unit Agents

These agents represent all those peripherals that help display and/or print results. They receive the petition from the computer and display, through whatever available means they have, some information to the user. Similar to the other agents, their methods are also mapped into memory by the computer agent using their manifest.

5 Simulation Building

One observation that the current document may get is the fact that it is proposed as a simulation instead of implementing the proposed architecture in a real computing system. However, a lot of work has been devoted in order to ensure that the simulation is as close as possible to a real system.

There are two components to consider here: the platform that enables agent-based implementation, and a way to generate programs that are significant in the sense that they require memory allocation, etc. Moreover, this programs

should at least be Turing-complete, meaning that it can compute every Turing-computable function [9].

The next lines of this section present both components. First, a description of the platform used to implement the agents and, second, the programs that were created to simulate real computing processes.

5.1 The MadKit Platform

MadKit is a multi-agent platform for developing and running application based on an organizational oriented paradigm. This multi-agent paradigms uses agents, groups and roles as the basic standpoint for building complex applications. MadKit does not enforce any consideration about the internal structure of agents, thus allowing a developer to freely implements its own agent architectures [10].

MadKit is also a distributed platform which allows for the development of efficient distributed applications. For the programmers, all considerations about basic distributed components such as "sockets" and "ports", are totally transparent [10]. An application developed in a multi-agent way can be run in a distributed way without changing a line of code.

MadKit is built around the concept of "micro-kernel" and agentification of services. The MadKit kernel is rather small, but agents offer the important services one needs for his own application. Distribution and remote message passing, monitoring and observation of agents, edition, etc. are all performed by agents [10].

5.2 HAL Programming Language

It was decided to create a new programming language that could allow the building of testing programs and, in that way simulate a computing system. This language was called HAL and it is Turing-complete [9]. It has most of the functions any high level programming language should have.

Anyway, a program coded is not sufficient as it needs a compiler, something that can transform a simple text with commands into something that is *executable*. Moreover, as the simulation is built in Java, it should provide a Java-based code that can be interpreted by the JVM. The next lines present both, the tool used to compile the HAL programs and the HAL programs themselves.

The Java Compiler Compiler. The Java Compiler Compiler (or JavaCC) is a parser generator and a lexical analyzer generator. Compilers and interpreters incorporate lexical analysers and parsers to decipher files containing programs, however lexical analysers and parsers can be used in a wide variety of other applications [11]. Lexical analysers can break a sequence of characters into subsequences called tokens and they also classify the tokens [11].

Usually, in compilers, the parser outputs a tree representing the structure of the program. This tree then serves as an input to components of the compiler responsible for analysis and code generation [11]. The lexical analyser and parser

are responsible for generating error messages, if the input does not conform to the lexical or syntactic rules of the language [11].

A part of the specification file used to create the HAL programs is shown lines below where mainly a few tokens are specified.

```
TOKEN :
{
  < ELSE: "else" >
| < FOR: "for" >
}
HALProgram Parse():
{Function main;
HALProgram result;}
```

HAL Program Characteristics. The HAL programming language comes with the required tools to program almost any kind of applications. It also offers the necessary *functions* for the simulation of the autonomic system. It is not object oriented but a functional program. It is capable to handle *String* and *Char* data types by mapping them as integers to memory. It is capable to declare arrays by typing `matrix = <5>;` or `->matrix = (10 11 12 13 14);`. In the first case, it allocates in memory an array of size 5 that is called `matrix`; in the second one, it already provides the elements to be assigned to the `matrix` variable. As it can be seen, the second example has at the beginning two special characters that are explained lines below.

Regarding the control statements, HAL provides *for*, *while*, and *if* commands. It also provides some already built-in methods for addition *sum*, subtraction *sub*, multiplication *mul*, division *div*, relational operators, etc. These methods have to be preceded by the at @ symbol so that JavaCC regards them as special tokens.

When handling values and variables, there are special characters that provide different data management. These special sequences of characters are:

Assign value to a memory address (− >). This sequence is used when one wants to assign a value to a specific memory address. It should always be on the left of an assignation (=). For instance, the line: `->30 = 4;` means that the value 4 should be stored in the memory address 30.

Get value from memory address (∗). This character gets the value that a specific memory address holds. For instance `*30` asks for the value stored in the memory address 30. If the previous example is considered, the result of using this character is 4.

Get a parameter from function (#). This one is used to get the parameters of a function. For instance if the function is `@sub(20 15);`, which asks to subtract 15 from 20, and then the assignation `paramOne = #1;` is called, the value that `paramOne` holds is 15. This operator is zero-based meaning that the first element is considered to be in the *zero* position.

One could say that these special characters work quite similar to the pointers C or C++ has.

As it can be seen, HAL offers all the possibilities to build sound programs. This characteristic is needed for the present simulation as one wants to test how the processor and memory agents behave as programs are executed following an agent-based

approach. For the purposes of the simulation many HAL programs have already been coded and are kept by the repository agent. It is wanted that these programs represent those that a user makes use of in real systems.

6 Testing the Model

In order to test the proposed model, a number of programs were generated and these were assigned a certain probability. These programs represent those a user make use of. Among these programs, some that attack the memory agent by erasing their restricted registries were coded. However, these agents have a very little probability of being generated. These can be considered as threats to the system. Then, the computer agent is started and it checks if there are sufficient resources to work. As previously mentioned, at least a processor and a memory agent should be present. Once these agents are acknowledged, the computer agent can star processing the programs that are being spawned by the repository agent. One also has the ability to add the components manually, kind of plugging in components and, as this is done, the system maps such new modules to memory.

The main idea behind this architecture is to enable self-⋆ properties so that autonomic computing is achieved. The next lines describe how such properties are enabled by the software architecture proposed here.

6.1 Self-⋆ Properties

Self-healing is provided here whenever an error is found. For the purposes of the simulation, there is a set of HAL programs that were created to disrupt the restricted memory sections were the registered modules have their methods and addresses mapped. These HAL programs change and erase the addresses and registries which in turn leads to an error. The memory agent monitors and ensures that no changes take place in the restricted area and, if such event does happen, it informs the computer of such occurrence. The computer agent then asks for the modules to send their manifests again and repairs the sections that the program might have damaged. The computer agent also, once the notification of an *attack* has been received, stops all programs from being spawned so that no more errors take place, and it only allows the communication of requests once the error has been fixed.

Finally, *self-protection* is achieved by the proposed architecture by not allowing the execution of known programs that *attacked* the system. Lines above, it was mentioned that some HAL programs were capable of damaging restricted memory, once such programs have been identified, the computer agent blacklists them and does not allow the instantiation of any more of those programs. This helps protect the system from known attacks.

In general, being this an agent-based approach, where all elements have been agentified, prevention and anticipation of events, optimization at any level of the system, and configuration based on the capabilities of each component are possible. The software architecture here proposed allows these features.

7 Conclusions & Final Remarks

Tackling autonomic computing by using an agent-based approach is a good path to follow. Agents have the ability to provide easy-to-implement self-⋆ properties as they rep-

resent a unit with their own methods, abilities and responsibilities. This document proposes an agent-based system architecture that enables the emergence of self-protection and self-healing properties, and makes use of a simulation to test the design.

This document tested the architecture using a simulation that allowed agent implementation (i.e., a virtual machine over the JVM). Moreover, in order to make it as real as possible, HAL programming language was developed using JavaCC [11]. Programs built using HAL are Turing-complete, resembling those used by people everyday. The presented software architecture agentifies all its components and establishes that all communications should go through one agent, called *computer agent*. This allows a better control of the system and hides the methods that other modules may have reducing possible attacks to the elements.

Worth pointing out is that this document presents a work-in-progress report and that there are still some things to be worked on.

Acknowledgments. This research project is funded by Tecnológico de Monterrey, Research Chair CAT010. The first author also thanks Jesús Héctor Domínguez Sánchez for providing insightful ideas and recommendations when designing this simulation.

References

1. Murch, R.: Autonomic Computing. On Demand Series. IBM Press (2004)
2. Parashar, M., Hariri, S.: Autonomic Computing: Concepts, Infrastructure, and Applications. CRC Press (2007)
3. Jennings, N.R.: On agent-based software engineering. Artificial Intelligence **117** (April 2000) 277 – 296
4. Tesauro, G., Chess, D.M., Walsh, W.E., Das, R., Segal, A., Whalley, I., Kephart, J.O., White, S.R.: A multi-agent systems approach to autonomic computing. In: AAMAS '04: Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems, IEEE Computer Society (2004) 464 – 471
5. Bonino, D., Bosca, A., Corno, F.: An agent based autonomic semantic platform. In: Proceedings of the International Conference on Autonomic Computing, IEEE Computer Society (2004) 189 – 196
6. Sterritta, R., Parasharb, M., Tianfield, H., Unland, R.: A concise introduction to autonomic computing. Advanced Engineering Informatics **19**(3) (July 2005) 181 – 187
7. Kephart, J.O., Chess, D.M.: The vision of autonomic computing. Computer **36**(1) (2003)
8. Blickle, T., Thiele, L.: A comparison of selection schemes used in genetic algorithms. Technical Report 11, Computer Engineering and Communication Networks Lab (TIK), Swiss Federal Institute of Technology (ETH) Zurich (1995)
9. Herken, R.: The Universal Turing Machine: A Half-Century Survey. Springer (1995)
10. MadKit: The MadKit project. `http://www.madkit.org/` (2005) Web Page.
11. Norvell, T.S.: The JavaCC Tutorial. `http://www.engr.mun.ca/~theo/JavaCC-Tutorial/javacc-tutorial.pdf` (2002)

Entropy-Based Profiles for Intrusion Detection in LAN Traffic

P. Velarde-Alvarado[1, 3], C. Vargas-Rosales[2], D. Torres-Román[1], and A. F. Martínez-Herrera[2]

[1]Department of Electrical Engineering and Computer Sciences,
Telecommunications Section, CINVESTAV-IPN,
Guadalajara, Jal., México
{pvelarde, dtorres}@gdl.cinvestav.mx
[2] Center of Electronics and Telecommunications
Instituto Tecnológico y de Estudios Superiores de Monterrey,
Monterrey, N.L., México
{cvargas, albertof_mtzherrera}@itesm.mx
[3] Department of Electronics,
Universidad Autónoma de Nayarit,
Tepic, Nay., México

Abstract. In this paper, a methodology for generating entropy-based behavior profiles of LAN traffic is proposed. The empirical analysis of our profiles through the rate of remaining features at the packet-level, as well as the three-dimensional spaces of entropy at the flow-level, provide a fast detection of intrusions caused by port scanning and worm attacks.

1 Introduction

Intrusion Detection Systems [1], or IDSs, have become an important component to detecting attacks against information systems. However, they offer only a limited defense. For instance, a signature-based IDS monitors packets on the network and compares them against a database of signatures or attributes from known malicious threats. A weakness of this type of IDS is that there will always be a lag between a new threat being discovered and the signature for detecting that threat being applied to the IDS. During that lag time the IDS would be unable to detect the new threat.

A second type of IDS is the anomaly-based IDS [2], which monitors network traffic and compares it against an established baseline. The baseline helps to identify what is normal behavior for that network. If a deviation from the established baseline reaches a specified threshold, an alarm is generated. Therefore, anomaly detection techniques have the potential to detect new and unforeseen types of attacks. Traditional anomaly based IDSs, employ algorithms that focus primarily on changes in the traffic volume at specific points on the network, and promptly alert the operator of a sudden increase.

© G. Sidorov (Ed.)
Advances in Artificial Intelligence: Algorithms and Applications
Research in Computing Science 40, 2008, pp. 119-130

However, such systems can be evaded through sophisticated attacks that focus on compromising significant hosts, causing them a collapse of memory or CPU and maintaining a level of traffic within the normal threshold.

Recently, a new generation of anomaly based IDSs have emerged, which focus on gaining knowledge in the structure and composition of the traffic and not just its volume. Such systems are based on the fact that the malicious activities affect the natural randomness of the network, e.g., they change significantly the entropy of the network [4]. The composition of traffic is related to its probability distribution, and can be characterized by its entropy] a malicious activity changes that composition and the shape of the distribution and therefore its entropy. By means of entropy measures to a set of traffic features, we can establish the profiles of normal activity of the network and determine intrusions to the system.

This paper presents an analysis at the packet and the flow level on traces obtained through measurements conducted in a campus network under real attacks of the Blaster [6] and Sasser [7] worms, as well as a port scan attack to the proxy server of that network. The captured traces during a week of normal operation, helped to develop a profile of normal behavior that is useful to be compared to attack conditions.

The paper is organized as follows. In section 2, we present our profiling approach and the context of this paper. Section 3 describes the test environment] section 4 and 5 explain the methodology: the rate of remnant items and spaces of entropy and results. Section 6 gives concluding remarks.

2 Profiling Approach

We propose two methods for the creation of profiles based on entropy. The analysis applies primarily to the packet-level for the method of the rate of remnant elements and to the flow-level for the spaces of entropy. Figure 1 shows the overall scenario, and this work is delimited by the gray box. Initially, there is a set of captured traffic traces corresponding to five days in typical work hours in an academic LAN. The traces have been inspected to be considered free of anomalies, so they may serve as a baseline.

We use traffic features to build the profiles. A traffic feature is a field in a header of a packet (at the packet level) or a field in a five-tuple (at the flow level), respectively. Four fields will be used: source address (*srcIP*), destination address (*dstIP*), source port (*srcPrt*), and destination port (*dstPrt*).

After the feature extraction, an essential part in the builder profile block is the measurement of entropy. For a discrete set of symbols $\{a_1, a_2, a_3, \cdots, a_n\}$ with probabilities p_i, i=1, 2,..., n, the entropy of the discrete distribution of a random variable X associated, is a measure of randomness in the set of symbols and represented as

$$H(X) = \sum_{i=1}^{n} p_i \log_2 p_i, \quad 0 \le H(X) \le H_{MAX} = \log_2 n. \tag{1}$$

The relative uncertainty (RU) provides a measure of variety or uniformity that is independent of the sample size. For a random variable X RU is defined as, [3],

$$RU = \frac{H(X)}{H_{MAX}}, \quad 0 \le RU(X) \le 1. \tag{2}$$

$RU(X) \approx 1$ means that observed values of X are closer to being uniformly distributed, thus less distinguishable from each other, whereas $RU(X) \approx 0$ indicates that the distribution is highly concentrated.

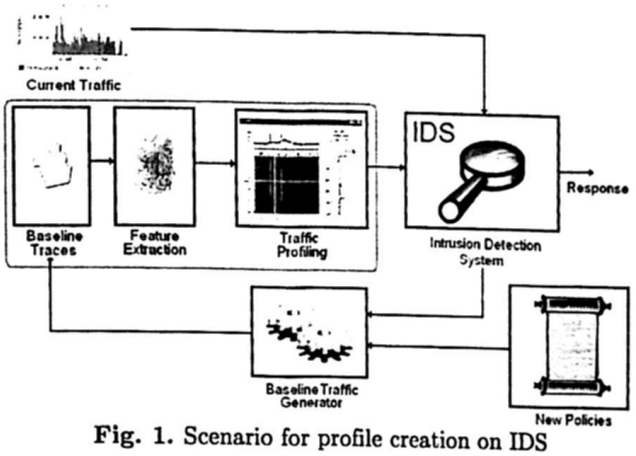


Fig. 1. Scenario for profile creation on IDS

3 Experimental Platform

The worm propagation and port scanning were carried out on academic LAN which is subdivided into four subnets (192.168.1.0, 192.168.2.0, 192.168.4.0, and 10.253.253.0). There are 100 hosts running Windows XP SP2 mainly. One router (192.168.1.1) connects the subnets with 10 Ethernet switches and 18 IEEE 802.11b/g wireless access points. The data rate of the core network is 100Mbps. A sector of the network is left vulnerable for worm propagation, with ten not patched Windows XP stations (192.168.1.104 – 113). In the experiments Blaster and Sasser worms where released in the vulnerable sector. The scanning port attack was observed on the proxy server (192.168.4.253).

3.1 Data set and tools

The benign traffic traces in typical work hours for a period of five days were labeled with a number from 1 to 5. The anomalous traffic for port scanning attack was labeled as 6-P1. Blaster and Sasser worm attacks were labeled as 6-P2 and 6-P4, respectively. The data-set was collected by a network sniffer tool based on *libpcap* library used by *tcpdump*, [8]. All traces were cleaned to remove spurious data using *plab*, a platform for packet capture and analysis, [9]. Traces were split into segments using *tracesplit* which is a tool that belongs to *Libtrace*, [10]. The traffic-files in ASCII format suitable for MATLAB® processing were created with *ipsumdump*, [11]. The flow generation was done with *flowanalyzer*, a tool based on *perl* and developed by us.

4 Rate of Remnant Elements

We base the methodology on the mathematical abstraction presented in our previous work [5], we define a traffic trace χ of a duration t_D seconds with a total of N packets, χ is divided into M non-overlapping slots of $t_d = \frac{t_D}{M}$ seconds each one. The i-th slot has W_i packets for $i = 1, 2, \cdots, M$. In each i-slot, four features are extracted that we associate with a value of r, namely $r = 1$ for source IP address, $r = 2$ for destination IP address, $r = 3$ for source TCP port, and $r = 4$ for destination TCP port. Let $\boldsymbol{S}$ be a finite sequence of $r = 1$ values or IP source addresses in a slot-i. This sequence with elements in an alphabet set A, is a function from $\{1, 2, 3, \cdots, |A|\}$ to A for some $|A| \geq 0$. The generated sequence $\boldsymbol{S}$ is denoted by $(a_1, a_2, a_3, \cdots, a_{W_i})$, and the length of $\boldsymbol{S}$ is W_i. The elements of $\boldsymbol{S}$ belong to an alphabet A with cardinality $n = |A|$. From A an ordered set $A^{(o)} = \{a_1^{(o)}, a_2^{(o)}, ..., a_n^{(o)}\}$ is created, $A^{(o)}$ contains the n-source IP addresses in decreasing order sorted by frequency. With the associated frequencies of A, we define a probability mass function (pmf)

$$\Pr\left(X_i^{r=1}, j\right) = p_j\left(a_j^{(o)}\right) = \begin{cases} f_j & 1 \leq j \leq n \\ 0 & rest \end{cases}, \tag{3}$$

where $f_1 \geq f_2 \geq f_3 \geq \cdots \geq f_n$. Ordered set $A^{(o)}$ is transferred to an iterative process Π to create l subsets of $A^{(o)}$ denoted as $A^{(o,k)}$, $1 \leq k \leq l$. This family of l subsets is shown in (4-6) and holds $A^{(o,k)} \setminus A^{(o,k+1)} = \{a_k^{(o)}\}$

$$A^{(o)} = A^{(o,1)} = \{a_1^{(o)}, a_2^{(o)}, a_3^{(o)}, ..., a_n^{(o)}\}, \tag{4}$$

$$A^{(o,2)} = \left\{a_2^{(o)}, a_3^{(o)}, a_4^{(o)}, \cdots, a_n^{(o)}\right\}, \tag{5}$$

$$\vdots$$

$$A^{(o,l)} = \left\{a_l^{(o)}, a_{l+1}^{(o)}, a_{l+2}^{(o)}, \ldots, a_n^{(o)}\right\}. \tag{6}$$

When in a k-iteration, the relative uncertainty of a partial pmf reaches a threshold β, i.e., $RU(X_i^r, k) > \beta$, we say that the iterative process Π reached its latest iteration, and hence, $k = l$. An estimator of relative uncertainty for a discrete random variable X_i^r in the k-iteration is defined in terms of its partial pmf as:

$$RU\left(X_i^r, k\right) = \frac{\hat{H}(P(X_i^r))}{\hat{H}_{MAX}} = \frac{\sum_{j=k}^{n} p_j(a_j^{(o)}) \log_2 p_j(a_j^{(o)})}{\log_2(n-k)} = \frac{\sum_{j=k}^{n} f_j \log_2 f_j}{\log_2(n-k)}, \tag{7}$$

Selecting a $\beta \approx 1$, the resultant subset $A^{(o,l)}$ is closer to being uniformly distributed. Then, for a given β, and a number l of iterations carried out, it is possible to calculate the remnant R_i^r for a subset $A^{(o,l)}$. Generalizing this for an i-slot and a r-traffic feature we have the rate of remnant elements:

$$R_i^r = \begin{cases} n & when\ all\ p_j(a_j^{(o)}) = \frac{1}{n}, n \geq 1 \\ n-l & for\ l \geq 1 \end{cases}. \tag{8}$$

In other words, R_i^r is the cardinality of the subset $A^{(o,l)}$. We found that this feature under normal conditions presents regularities that allow creating behavioral traffic profiles. Table 1 summarizes the R_i^r behavior with $\beta = 0.95$ for our data-set.

Through of mean, variance, the intensity factor ($\frac{\sigma^2}{\mu}$), and maximum value we can define a threshold for normal behavior of R_i^r. For instance, by averaging the means of R_i^r and its maximum values during benign traffic, we can define an average threshold of 28.5 with a maximum of 114.8 units. We denoted these thresholds for each r by $T(R_i^1) = (28.5; 114.8)$, $T(R_i^2) = (31.7; 115.6)$, $T(R_i^3) = (92.0; 342.6)$, and $T(R_i^4) = (132.3; 542.2)$.

Figure 2 shows the four patterns R_i^r for benign traffic in Trace 5 and its variation is inside of standard behavior for R_i^r.

Table 1. Values of mean, variance, and intensity factor for the rate of remnants

Trace	Mean				Variance				Intensity Factor			
	srcIP	*dstIP*	*srcPrt*	*dstPrt*	*srcIP*	*dstIP*	*srcPrt*	*dstPrt*	*srcIP*	*dstIP*	*srcPrt*	*dstPrt*
1	29.1	33.1	103.5	136.5	328.2	377.5	4,502	8,818	11.29	11.42	43.5	64.61
2	29.5	35.1	88.8	138.6	566.5	688.1	3,830	12,918	19.23	19.62	43.16	93.24
3	31.1	33.0	96.2	141.7	497.9	595.2	4,598	11,972	15.99	18.06	47.78	84.51
4	31.1	35.9	103.2	141.9	732.8	727.9	6,209	18,868	23.59	20.26	60.16	133
5	21.5	21.5	69.0	102.88	277.8	351.1	3,208	9,191	12.94	16.33	46.5	89.3
6-P1	14.5	16.6	76.1	74.5	145.1	166.8	4,916	4,958	10.0	10.05	64.6	66.53
6-P2	27.9	19,618	3,107	1,005	3,464	6.2c07	1.7c06	4.3c04	12.42	3,209	549.7	43.24
6-P4	3,149	5,214	3,045	2.243	97,058	1.2e06	2.4e05	2.4e05	310.9	237.5	90.14	111.24

An anomaly related with a port scan attack directed to the proxy server was possible to detect it since the first slot that appeared (i.e i=2, 3). The attack was carried out across a large number of TCP packets with source addresses supplanted. The growth of R_i^1 is possible to observe in Figure 3 and is far away from $T(R_i^1) = (28.5;114.8)$.

R_i^r patterns during worms attacks are presented in Figures 4 and 5. There is an important grown for $r = 2,3,4$ for Blaster Worm and for all R_i^r during Sasser Worm attack. It is important to note that the anomaly detection is done from the earliest slots that the intrusion appears.

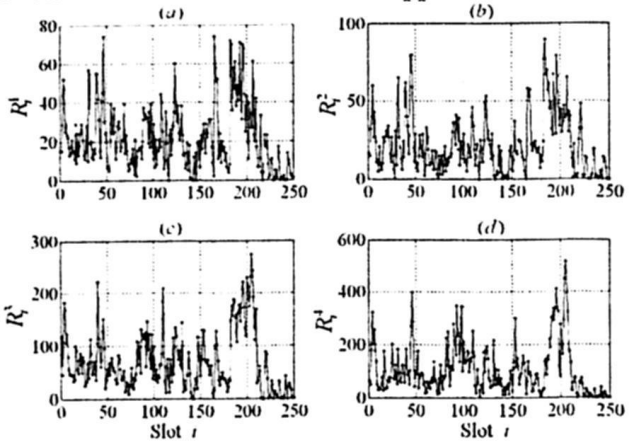


Fig. 2. Rate of remnant for (a) *srcIP*, (b) *dstIP*, (c) *srcPrt* and (d) *dstPrt* for standard traffic in Trace 5 in typical work hours. ($t_d = 60s$, and $\beta = 0.95$)

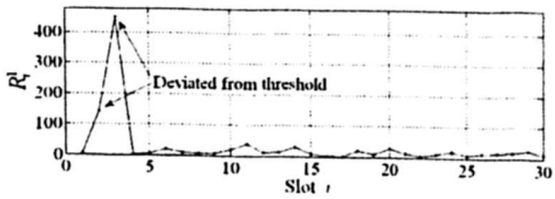


Fig. 3. Rate of remnant for *srcIP* ($r=1$) under port scan attack using spoofed IP addresses which is observable in slots i=2 and i=3 on Trace 6-P1 ($t_d = 60s$, and $\beta = 0.95$)

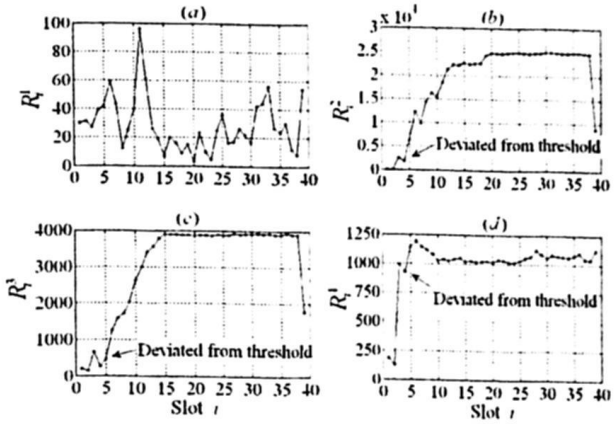


Fig. 4. Rate of remnant for (a) *srcIP*, (b) *dstIP*, (c) *srcPrt* and (d) *dstPrt* during Blaster Worm on Trace 6-P2 ($t_d = 60s$ and $\beta = 0.95$)

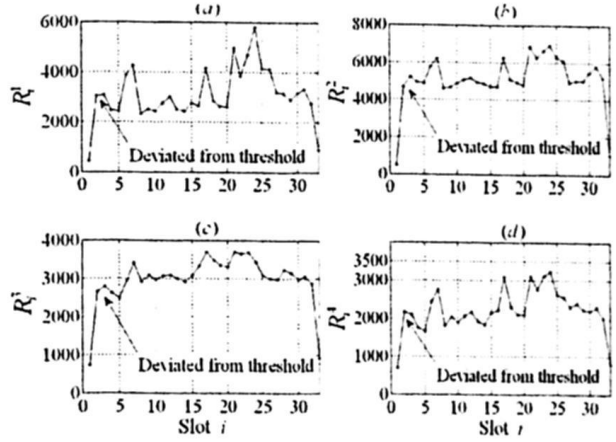


Fig. 5. Rate of remnant for (a) *srcIP*, (b) *dstIP*, (c) *srcPrt* and (d) *dstPrt* during Sasser Worm on Trace 6-P4 ($t_d = 60s$ and $\beta = 0.95$)

5 Three-Dimensional Spaces of Entropy

The construction of a space of entropy is carried out at flow level, and through these spaces is possible to create profiles of behavior for the traffic of a network. Four three-dimensional spaces are generated for each one of the features extracted from the flows. We define a traffic trace χ of a duration t_D seconds that is divided into M non-overlapping slots of $t_d = \frac{t_D}{M}$ seconds each one. In an i-slot K_i flows are generated with a given inter-flow gap (IFG). All the flows for each slot are stored on indexed text files. The traffic features used in this technique are the flow's fields and are identified as $r=1$ for source IP address, $r=2$ for destination IP address, $r=3$ for source TCP port, and $r=4$ for destination TCP port.

Once that flows in an i-slot are generated, they should be clustering according to a r-flow feature. For instance, with a cluster key or pivot $r=1$ the flows are aggregated into those flows that share the same source IP address. The number of clusters depends on $\left|A_i^{r=1}\right|$, where $A_i^{r=1}$ is the alphabet set of all source IP addresses seen in the slot i. Thus, each cluster has flows with the same source IP address, but the rest of fields or features ($r=2,3,4$) have freedom of variation. In this context, we can estimate the entropy for each $r=2,3,4$ of each cluster. If we join these three values and associate them with a coordinate, we have a cloud of data points in a 3-D Euclidean space, where the axis are $(\hat{H}_{\text{srcPrt}}, \hat{H}_{\text{dstPrt}}, \hat{H}_{\text{dstIP}})$ for $r=1$. Finally, the $\left|A_i^{r=1}\right|$ points in the slot i, that is, $(\hat{H}_{\text{srcPrt}}, \hat{H}_{\text{dstPrt}}, \hat{H}_{\text{dstIP}})_1, (\hat{H}_{\text{srcPrt}}, \hat{H}_{\text{dstPrt}}, \hat{H}_{\text{dstIP}})_2$, ... $(\hat{H}_{\text{srcPrt}}, \hat{H}_{\text{dstPrt}}, \hat{H}_{\text{dstIP}})_{\left|A_i^{r=1}\right|}$ are plotted in the 3D-space. When we apply this procedure to the rest of cluster keys and all slots, we get four spaces of entropy.

Figures 6, 7 and 8 show the spaces of entropy for traces with $t_D = 38\,\text{min}$. First, in Figure 6, we see the shape for Trace-1, which corresponds to normal traffic conditions being typical for Traces 2 - 5. Figures 7 and 8 show a marked difference with regard to benign traffic, since the data points move away from positions typically observed.

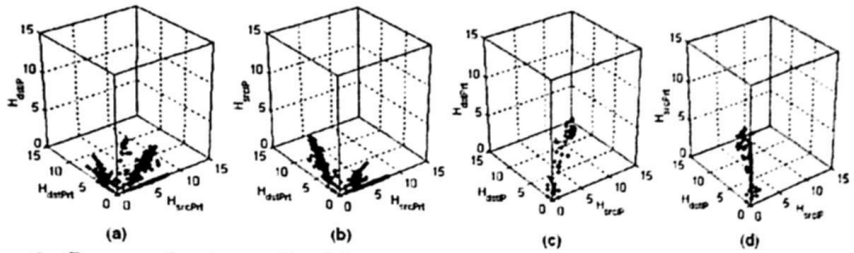


Fig. 6. Spaces of entropy for (a) srcIP cluster key, (b) dstIP cluster key (c) srcPrt cluster key, and dstPrt cluster key for traffic Trace -1 in typical work hours for a 38 min period

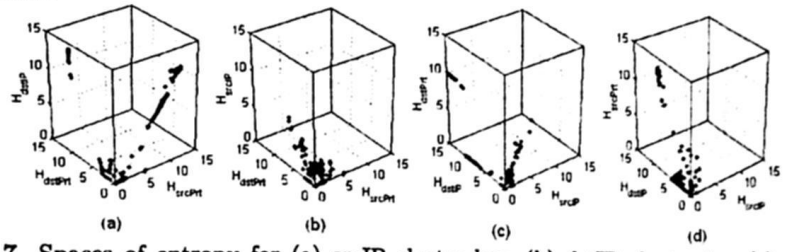


Fig. 7. Spaces of entropy for (a) srcIP cluster key, (b) dstIP cluster key (c) srcPrt cluster key, and dstPrt cluster key for anomalous traffic Trace 6-P2 (Blaster Worm) during 38 min period

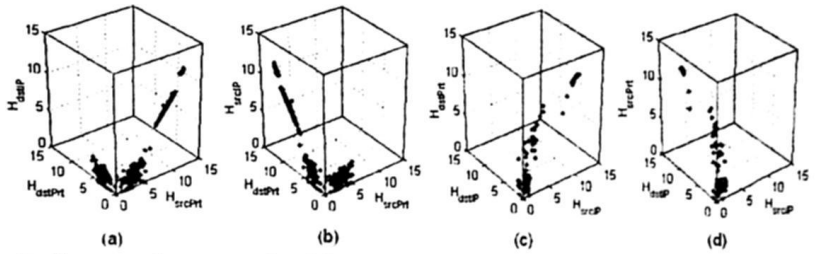


Fig. 8. Spaces of entropy for (a) srcIP cluster key, (b) dstIP cluster key (c) srcPrt cluster key, and dstPrt cluster key for anomalous traffic Trace 6-P4 (Sasser Worm) during 38 min period

The characterization of the spaces of entropy represented by the vector $\mathbf{X}^r \in \square^3$ for a cluster key r was realized applying initially a technique of multivariable analysis, the Principal Component Analysis. PCA provides a roadmap for how reduce a complex data-set to a lower dimension $\mathbf{Z}^r \in \square^d, d \leq 3$ to reveal the sometimes hidden, simplified structure that often underlie it. PCA is mathematically defined [12] as an orthogonal linear transformation that transforms the data to a new coordinate system such that the greatest variance by any projection of the data comes to lie on the first coordinate (called the first principal component, PCA 1), the second greatest variance on the second coordinate, and so on.

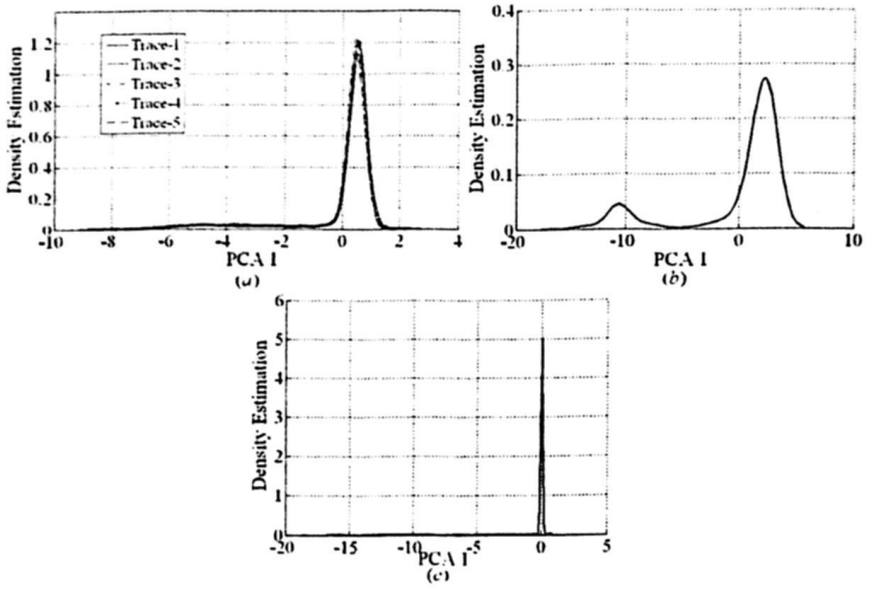


Fig. 9. Kernel density estimation for $\mathbf{Z}^{r=1}$ in (a) Trace 1-5 (Benign Traffic), (b) Trace 6-P2 (Blaster attack), and (c) Trace 6-P4 (Sasser attack)

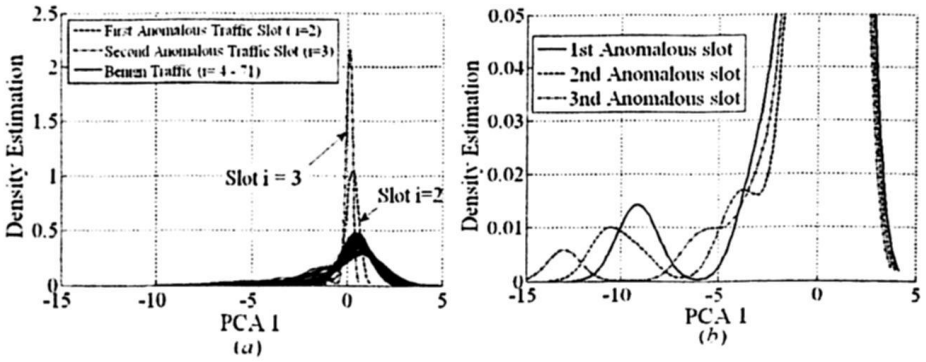


Fig. 10. Anomaly Detection on $\mathbf{Z}_i^r$ (a) Anomaly caused by port scan detected in first slot in Trace 6-P1, (b) Anomaly caused by the worm deviates the second mode out of the threshold

The evaluation of the behavior of the Principal Component for the transformed space $\mathbf{Z}^{r=1}$ was observed by means of its estimated probability density, with KDE (Kernel Density Estimation) using a 200 points-Gaussian Kernel, and a bandwidth $h = 1.06\sigma n^{1/5}$. In Figure 9a shows that the densities of the transformed spaces (i.e., $\mathbf{Z}^{r=1}$) of the benign traffic traces, present a clear regularity in its form. This pattern of behavior changes drastically for traces of anomalous traffic (Figure 9b and 9c). This procedure was applied to

slots i for a given r, now the transformed space is denoted as $\mathbf{Z}_i^r$. Densities of $\mathbf{Z}_i^r$ under normal conditions presents bimodality, the second mode is situated on the left side of the main mode, at an average of -5 units. The average variance of $\mathbf{Z}_i^r$ is 3 units. The values of variance for $\mathbf{Z}_{i-2}^{r-1}$ and $\mathbf{Z}_{i-3}^{r-1}$ are 0.74 and 0.32, respectively, and represent an anomaly with regard to the threshold of 3 units. Thus, an intrusion (Figure 10a) is early detected in Trace 6-P1. In the Figure 10b the three first slots with the attack cause that the second mode displace to -9, -11 and -13, representing an anomaly with regard to the threshold of -5.

6 Conclusions and Future Work

The generation of behavioral profiles based on entropy offers an effective support for the Intrusion Detection Systems. The results of this study in a campus network show that under the Blaster and Sasser worm attacks as well as the port scanning, an IDS employing profiles generated by the Rate of remnant elements or Three-Dimensional Spaces of Entropy methodologies can provide a rapid response detecting deviations from an established baseline in the early slots that the attack appears.

As a future work, we will investigate the effect on variation of the slot duration t_d, smaller values of slot duration represent faster response times, but also represent a smaller data set where to obtain representative traffic features, finding the optimum value is an important objective design.

References

1. Bolzoni, D., Etalle, S.: Approaches in Anomaly-based Network Intrusion Detection Systems. Intrusion Detection Systems: Advances in Information Security. Springer Science+Business Media, LLC (2008)
2. Kruegel, C., Valeur, F. Vigna, G.: Intrusion Detection and Correlation. Advances in Information Security. Springer (2005)
3. Xu, K., Zhang, Z., Bhattacharyya, S.: Profiling Internet Backbone Traffic: Behavior Models and Applications. SIGCOMM 2005. (2005) 22-26
4. Nucci, A., Bannerman, S.: Controlled Chaos. IEEE Spectrum. Vol.44. No.12. (2007) 42-48
5. Velarde-Alvarado, P., Vargas-Rosales C., Torres-Roman D., Munoz-Rodriguez, D.: Entropy Based Analysis of Worm Attacks in a Local Network. Research in Computing Science. Vol. 34, (2008) 225-235
6. Copley, D., Hassell, R., Jack, B., Lynn, K., Permeh, R., Soeder, D.: ANALYSIS: Blaster Worm. eEye Digital Security Research. http://research.eeye.com/html/advisories/published/AL20030811.html
7. Ukai, Y., Soeder, D.: ANALYSIS: Sasser. eEye Digital Security Research. http://research.eeye.com/html/advisories/published/AD20040501.html

8. Jacobson, V., Leres, C., McCanne, S.: Tcpdump/libpcap. http://www.tcpdump.org/
9. Peppo, A. plab. Tool for traffic traces. http://www.grid.unina.it/software/Plab/
10. Trac Project. Libtrace. http://www.wand.net.nz/trac/libtrace
11. Kohler, E. ipsumdump. Traffic tool. http://www.cs.ucla.edu/~kohler/ipsumdump
12. Jolliffe I.T.: Principal Component Analysis, Series: Springer Series in Statistics, 2nd ed., Springer, (2002), XXIX, 487 p. 28

Intrusion Detection for Mobile Ad-Hoc Networks based on a Non-Negative Matrix Factorization Method

Carlos Mex-Perera[1] José Zamora-Elizondo[1] Raul Monroy[2]

[1] Center for Electronics and Telecomunications, ITESM, Campus Monterrey
Av. Eugenio garza Sada 2501 Sur. Col. Tecnologico
Monterrey, N.L. , CP 64849 Mexico
[2] Computer Science Department, ITESM, Campus Estado de Mexico
Carretera al lago de Guadalupe, Km. 3.5, Estado de Mexico, CP 52926, Mexico
{carlosmex, A00791990, raulm}@itesm.mx

Abstract. In this paper we focus on intrusion detection in Mobile Ad Hoc Networks (MANETs), we propose a novel method for intrusion classification based on a Non-Negative Matrix Factorization (NMF) model. Feature vectors derived from statistics collected in the routing tables of the mobile nodes are used to form an input matrix for the NMF algorithm, which creates a behavior profile by building a matrix W of basis. Such matrix is later used to test unseen vectors. The distance between the test vector and its reconstruction is compared to a threshold level to obtain a decision about the existence of normal behavior. The results of the simulations show that the method might be suitable for its deployment in MANETs.

1 Introduction

A Mobile Ad hoc Network (MANET) is a low-cost, rapid-deployment, self-configuring network of mobile nodes (and associated hosts) connected by wireless links. Nodes cooperate one another forwarding packets so that each node may communicate beyond its wireless transmission range. They are free to move forming an arbitrary topology, which changes rapidly and unpredictably. MANETs have captured increasing interest as they are suitable for emergency situations.

Security is required in many MANET applications, including military operations and disaster relief. However, MANETs are vulnerable to a number of attacks. At a communication level, an intruder can easily inject bogus packets or eavesdrop on communication. At a network level, an intruder can easily attempt a malicious router misdirection. MANET's vulnerabilities cannot always be dealt with using techniques that were designed in the context of wired networks [1]. This is particularly the case for ad hoc routing: the routing problem is magnified as soon as we no longer assume a trusted environment [2]. This is because it is not easy to distinguish an ordinary change in the network topology from a change caused by a collection of compromised nodes.

Advances in Artificial Intelligence: Algorithms and Applications
Research in Computing Science 40, 2008, pp. 131-140

Security mechanisms, such as authentication and encryption can be used as the first line of defense against attacks in MANET. However, they still cannot provide protection for attacks generated by a malicious inside node. Intrusion detection mechanisms are necessary to detect this type of attacks. To mitigate the problem, Intrusion Detection Systems (IDS), as a complementary mechanism, is designed to protect the availability, confidentiality and integrity of critical networked information systems. The goal of a IDS is to distinguish those nodes that perform an attack, such nodes are known as intruders.

In recent years the problem of IDS for MANEts has been devoted an special interest, there are a number of papers that have proposed different mechanisms for IDS for MANET, such as [3–5].

In this paper, we focus on the problem of intrusion detection for MANETs in the network layer, we propose a IDS based on a Non-Negative Matrix Factorization [6] model. Although NMF has been used previously in [7], for profiling program and user behaviors for host-based IDS, it has not been applied yet in MANETs. This work presents results that show that NMF could be applied in MANETs.

2 Intrusion Detection

Intrusion detection is concerned with the timely discovery of any activity that jeopardizes the integrity, availability or the confidentiality of an IT system. A Misuse Intrusion Detection System (MIDS) annotates as an attack any known pattern of abuse. MIDSs are very effective in detecting known attacks but are usually bad at detecting novel attacks. An Anomaly IDS (AIDS) annotates as an attack any activity that deviates from a profile of ordinary computer usage. Unlike MIDSs, AIDSs are capable of detecting novel attacks. However, they frequently tag ordinary computer usage as malicious, yielding a high false positive detection rate.

Depending on the activity it observes, an IDS can be placed at either of three points: a host, a network or an application. A host IDS usually audits the functionality of the underlying operating system, but can also be set to watch critical resources. An application IDS scrutinizes the behavior of an application. It commonly is designed to raise an alarm any one time the application executes a system call that does not belong to a pre-defined set of system calls, built by some means, an object-code analysis. A network IDS analyzes network traffic in order to detect mischievous activities within a computer network. A denial of service attack resulting from flooding a network with packets can be pinpointed only at this level.

An IDS should perform a number of tasks. In particular, it should [8]:

- identify the appearance of patterns of a known attack or of deviations from normal computer usage;
- identify flaws or vulnerabilities in the system configuration;
- audit the integrity of critical system or data files; and
- highlight user violations of a security policy.

Additionally, an ad hoc network IDS should [1]

- not add any extra weakness to the computer system under surveillance;
- consume little system resources; and
- run continuously in a transparent manner.

2.1 Building an Anomaly detection Model

This work is focused on obtaining profiles from statistics computed from the routing tables of the nodes of the MANET. Attacks can be presented in a great number of ways, for instance an intruder node can fake routing information and all nodes that receive such information might be building their routing tables with erroneous entries. Another way to create an attack is by dropping packets, thus the rest of the nodes would not be updating their routing information as expected in absence of intruders. In both type of attacks, connectivity among the nodes would be affected according to the severity of the attack. In our study, we implemented random packet dropping attack which is a pattern distortion technique.

In order to build an anomaly detection model, statistics of the routing tables when intruders are not present in the MANET are considered. Such statistics are then properly formatted to generate data vectors to train a classifier.

Once the learning phase has been performed, the classifier is ready to be used to test unseen vectors. The result of the classification is a decision about if the observed vector corresponds or not to a normal behavior. If an abnormal behavior is obtained, then it is considered that an intruder is affecting the routing protocol.

3 The Proposed Method

The proposal is based on non-negative matrix factorization, which is a method aimed to represent data using non-negativity constraints. The idea is the representation of a given object using the addition of its parts, which are considered as positive contributions to the whole object. NMF has been applied to many fields, including face recognition and text classification tasks [6]. Considering that intrusion activities in MANETs might affect audit data as positive contributions, we propose the use of NMF for intrusion detection.

3.1 Non-negative Matrix Factorization

Given a database represented by a n x m matrix V, where each column is an n-dimensional vector with positive data belonging to the original database (m vectors), we can obtain an approximation of the whole database (V) by

$$V_{i\mu} \approx (WH)_{i\mu} = \sum_{a=1}^{r} W_{ia} H_{a\mu} \quad (1)$$

Where the dimensions of the matrix W and H are n x r and r x m, respectively. Usually, r is chosen so that $(n+m)r < nm$. This results in a compressed version of the original data matrix. Each column of matrix W contains a basis vector while each column of H contains encoding coefficients needed to approximate the corresponding column in V. The following iterative learning rules are used to find the linear decomposition [6]:

$$H_{a\mu} \longleftarrow H_{a\mu} \sum_i (V_{i\mu}/(VH)_{i\mu}) W_{ia} \tag{2}$$

$$W_{ia} \longleftarrow W_{ia} \sum_\mu (V_{i\mu}/(WH)_{i\mu}) H_{a\mu} \tag{3}$$

$$W_{ia} \longleftarrow W_{ia}/\sum_j W_{ja} \tag{4}$$

The above unsupervised multiplicative learning rules are used iteratively to update W and H. The initial values of W and H are fixed randomly. With these iterative updates, the quality of the approximation of Equation 1 improves monotonically with a guaranteed convergence to a locally optimal matrix factorization [6].

3.2 IDS Based on Non-negative Matrix Factorization

IDS Based on NMF includes three steps: features selection, classifier training, classifier test and decision, as it is shown in Figure 1.

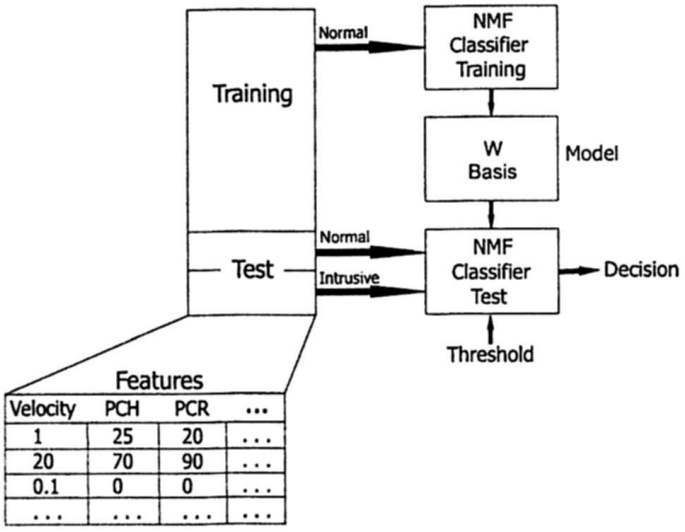


Fig. 1. System Architecture.

3.3 Features selection

Because we focus on routing attacks, we need to specify the trace data to be used that will present evidence of normalcy or anomaly. We define a trace data as the set of features aggregated into a single data set, which describes all changes in routing tables for a single node. Routing tables have information to know the next hop to reach a destination node and the number of hops for that route. Due to the movement of the nodes, the routing tables are updated regularly. These changes in the routing tables can be computed as features for a classifier. Following the work described in intrusion detection of MANETs [3], and based on our experiment results, we use features associated with routing caches and topological movement of mobile nodes in order to characterize their normal changes. Figure 1 shows some fictional features for a node. All features are detailed in Table 1 and the meaning of each feature is further explained in the Notes column.

Features	Explanation	Notes
PCR	% of changed routes	Deleted and increased routing entries
PCH	% of changes in the sum of hops	Average length of routes
PCB	% of change of bad routes	Broken routes
PCS	% of change of stale routes	Stale routes being removed
PCU	% of change of updated routes	Routes updated via overhearing.

Table 1. Local features ad-hoc routing protocol

We use percentages as measurements because of the dynamic nature of mobile networks (i.e., the number of nodes/routes is not fixed).

3.4 Classification Training

The data set for normal behavior represented by V (which is the matrix transpose of training data, see Figure 2) is approximately factorized into two matrices $W_{training}$ and H by the iterative updating rules given by Equations 2 - 4. Each column of matrix $W_{training}$ contains a basis vector, while each column of H represents the coefficients needed to approximate the corresponding column in V. Figure 2 shows that given a set of multivariate n-dimensional data vectors, the vectors are placed in the columns of an $n \times m$ matrix V where m is the number of examples in the data set. For instance, a column of matrix V contains a vector data with the values of PCR, PCH, PCU, PCB, PCS and velocity at a given time t, others columns of V have the same features but taken at different times. This matrix V is then approximately factorized in $W_{training}$ and H.

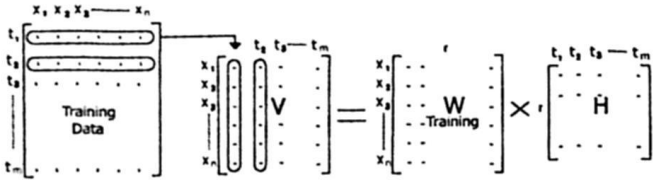


Fig. 2. NMF Classifier training.

3.5 Classification Test and decision

Equation 1 can be rewritten column by column as $v \approx Wh$ where v and h are the corresponding columns of V and H, respectively. Each vector v is approximated by a linear combination of the columns of W, weighted by the components of h. Based on this, it is found that given a column v of matrix V_{test} and using the basis $W_{training}$ learned from normal training data set, we can find a representative vector of encoding coefficients h by the update rule in Equation 2, see Figure 3. We then reconstruct v as $v' = W_{training} \times h$.

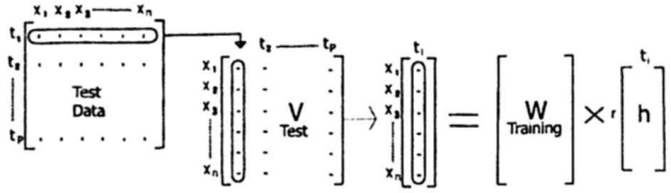


Fig. 3. NMF Classifier test.

The Euclidean distance given by Equation 5 between v and v' is used to calculate the similarity between these two vectors.

$$\Delta = \| v - v' \|^2 = norm(v - v') \tag{5}$$

If the test data vector is normal, then it is expected that the test data vector v will be very similar to its reconstructed version v', and the resulting distance between them will be very small. Therefore the testing vector is classified as normal if

$$\Delta < \varepsilon \tag{6}$$

where ε is defined as a threshold. Otherwise it is classified as intrusive, on this property our intrusion classification model is based.

4 Experimental Studies

In order to study how NMF classifier can be used to construct anomaly detection models for MANET routing, we have conducted the following simulation experiments.

4.1 Simulation Model

We chose one specific ad-hoc wireless protocol as the subject of our study, Ad-hoc On-Demand Distance Vector (AODV) [9]. In the simulation, 50 mobile nodes move in a 1000 X 1000 meter square region. We apply the random way-point model to emulate node mobility patterns. The maximum pause time between movements is 300 seconds, the minimal movement speed is 1 m/s, and the maximal movement speed is 20 m/s. 16 source-destination pairs are selected randomly to generate Constant Bit Rate (CBR) traffic as the background traffic. The transmission range is set to 250 meters. We simulate a routing disruption attack, where the attacker node drops packets belonging to the routing protocol, the attacker was randonmly chosen among 50 nodes. We run the simulation 3000 seconds in order to get normal data traces from all nodes. For each data trace, we collect (PCR, PCH, PCU, PCB, PCS and velocity) feature values every 3 seconds after a warm-up time period of 300 seconds. The data at the last 100 seconds are discarded. Values are treated in a similar way presented in previous works, in our case all features except the velocity are discretized into five uniformly distributed levels, ranging from 0 to 100% while velocity feature is discretized into 10 levels. We split the data collected during the 3000 seconds simulation into 2 parts, the first one corresponding to the first 2400 seconds will be used as training data and the rest 600 seconds will be used as normal testing data. Data collected from the routing tables of all nodes forms the training and testing vectors, each training data trace has 800 items and each normal testing data trace has 200 items. In this way, we get 50 x 800 = 40000 items for training data, and 50 x 200 = 10000 items for normal testing data. To get data of intrusive behaviors, we let the simulation run 600 seconds. For each run, we let the attack script start at time 100 seconds, and ends at time 300 seconds. We get 50 x 200 = 10000 items for intrusive testing data.

4.2 Simulations results

For evaluation purposes of the detection performance of the proposed method, we obtain Receiver Operating Characteristic (ROC) curves. An ROC curve is a parametric curve that is generated by varying the threshold of the intrusive measure, which is a tunable parameter. The ROC curve can be used to determine the performance of the system for different operating points. The ROC curve is the plot of False Alarm Rate, calculated as the percentage of decisions in which normal data are flagged as intrusive versus Missing Alarm Rate, calculated as the percentage of intrusive behavior falsely classified as normal. Training data and test data are used to tune the parameters of the classifier. The dimension r

is typically much lower then either dimension of matrix V. We have trained the model using different basis r from 2 to 5 in order to determine a suitable value of r. See Figure 4, among the ROC curves depicted the best results are obtained for $r = 3$. It is interesting to observe that NMF classifier has different operating points and we can be positioned in any point of the curve depending on the necessities. For comparison purposes we run the simulation with the same data using RIPPER [10], which is a well known rule based classifier. For RIPPER it is obtained a false alarm rate (1.67%) and the missing alarm rate (71.26%), which for many cases is not acceptable. Besides, it is not possible to adjust the sensibility of the detection for RIPPER. However, with the NMF method it is possible to adjust the operating point following a trade-off between the false alarm and missing alarm rates.

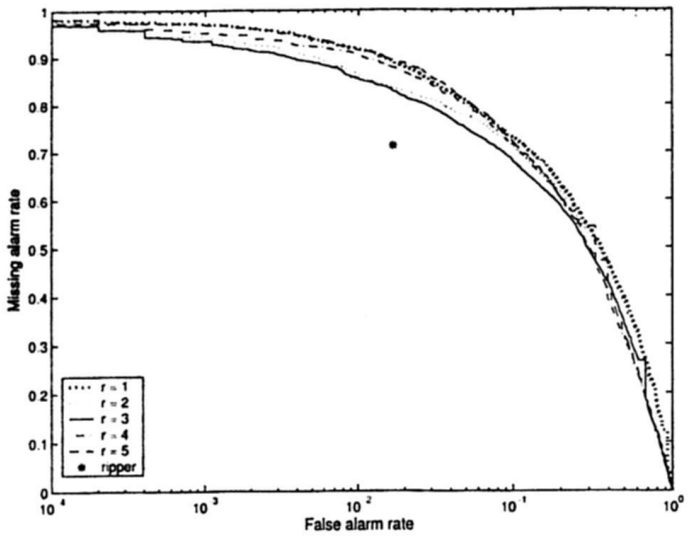


Fig. 4. ROC performance curves comparing NMF using from r = 2 to r = 5 basis and RIPPER

In our experiments, we performed for training 50 iterations (update rules) of the NMF algorithm while for testing phase we only performed 10 iterations. The reason for this reduction in the number of iterations in the test phase relies on the fact that the NMF algorithm converges quickly for normal test data while it does not converge if the test data is abnormal. Table 2 shows the training times of the classifier in seconds for different sizes of the training data. Despite the code for NMF was not optimized, it is observed that our model is faster compared to the classifier using RIPPER. In more detail, our model can learn from different size of normal training data in a short time even if the the amount of audit data is quite large. We measured the time to process test data of the proposed

method for several amounts of data, the results are listed in Table 3. From the results it can be concluded that performance of the NMF based method is very attractive for real time applications where actions for response must be taken as soon as possible.

Training Data	NMF	RIPPER
40000	7.0300	364.4
20000	3.1040	309.88
10000	2.1330	233.25
5000	1.4930	102.04
1000	0.9820	43.17
500	0.9310	30.34

Table 2. Learning time(seconds) NMF vs. RIPPER

NMF would be suitable for scenarios where computational resources are limited since the processing is very fast. Another advantage of using NMF is that if we want to update our model, we only need to update the matrix $W_{training}$, which results in a reduced computational cost.

Test Data	Processing time
10000	5.7480
5000	2.4130
1000	0.4400
500	0.2400

Table 3. Testing time(seconds) using NMF for different size of testing data

5 Conclusion

The experiment results demonstrate that Non-Negative Matrix Factorization might work on MANETs. Normal behavior profiles of a routing protocol can be established and used to detect anomalies. However, more research has to be done to obtain operating points with both low false and missing alarm rates. Thus, less error rates should be reached to meet the typical goals for a real IDS. It is suggested that the preprocessing stage should be improved for better performance of the detection methods. On the other hand, the model can easily achieve real-time intrusion classification based on dimensionality reduction and on a simple classifier. In the proposed method, NMF reduce the high dimensional

data vectors and classifies in a low dimensional space with high efficiency and low usage of system resources. The NMF algorithm does not seem sensible to parameter selection, furthermore a small amount of data is sufficient to train the classifier without reducing the performance. Updating the profiles can be easily implemented in the NMF model, a number of iterations of the algorithm can be used for updating purposes.

The low computational expense of the processing allows a real-time performance of intrusion classification. Future work might involve research considering more attack scenarios in MANETs, not only at the routing layer, but also at other layers. More security-related features can be drawn after the analysis of the threats. This could facilitate the construction of better detection models.

Acknowledgments. The authors would like to acknowledge the Cátedra de Biométricas y Proto colos Seguros, ITESM, Campus Monterrey, Cátedra de Seguridad, ITESM, Campus Estado de Mexico and Regional Fund for Digital Innovation in Latin America and the Caribbean (FRIDA), who supported this work.

References

1. P. Albers, O. Camp, J. M. Parcher, B. Jouga, L. Me, R. Puttini, "Security in Ad Hoc Networks: a General Intrusion Detection Architecture Enhancing Trust Based Approaches", The 1st International workshop on Wireless Information Systems" (WIS 2002), in the 4rth International Conference on Enterprise Information Systems, 2002.
2. W. Wang, Y. Lu, B. K. Bhargava, "On Vulnerability and Protection of Ad Hoc On-demand Distance Vector Protocol", International Conference on Telecommunications (ICT'2003), France, March 2003 pp. 375 - 382 Vol.1.
3. Y. Zhang and W. Lee, "Intrusion Detection In Wireless Ad-Hoc Networks", Proceedings of the 6th International Conference on Mobile Computing and Networking, MobiCom 2000, pp. 275-283, August 2000.
4. Hongmei Deng, Qing-An Zeng, Agrawal D.P., "SVM-based Intrusion Detection System for Wireless Ad Hoc Networks", Vehicular Technology Conference, 2003. VTC 2003-Fall. 2003 IEEE 58th, Volume 3, 6-9 Oct. 2003 pp. 2147 - 2151 Vol.3
5. Buschkes R., Kesdogan D., Reichl P., "How to increase security in mobile networks by anomaly detection", Computer Security Applications Conference, 1998, Proceedings., 14th Annual 7-11 Dec. 1998 pp. 3 - 12
6. D. D Lee and H Sebastian Seung, "Learning the parts of objects by non-negative matrix factorization", Nature, vol. 401, pp. 778-791, 1999.
7. Wei Wang, Xiaohong Guan, Xiangliang Zhang, "Profiling program and user behaviors for anomaly intrusion detection based on non-negative matrix factorization", Decision and Control, 2004. CDC. 43rd IEEE Conference on Volume 1, 14-17 Dec. 2004 pp. 99 - 104 Vol.1
8. H. Debar, M. Dacier, A. Wespi, "Towards a taxonomy of intrusion-detection systems", Computer Networks 31, 1999, pp. 805-822.
9. S. Marti, T.J. Giuli, K. Lai and M. Baker, "Mitigating Routing Misbehavior in Mobile Ad Hoc Networks", Proceedings of MobiCom 2000, pp. 255-265.
10. W.W. Cohen, "Fast effective rule induction", in: Proceedings of the 12th International Conference on Machine Learning (Morgan Kaufmann, San Mateo, CA, 1995) pp. 115-123.

Author Index

Aréchiga, MA 49
Bautista-Thompson, Ernesto 3
Carrillo, Humberto 13
Cervera , Alejandra 27
Cortés Pérez, Ernesto 95
De la Cruz-De la Cruz, Luis 3
Felipe-Riveron, Edgardo 83
Fernández-Carrasco, Luis M. 109
García Aguilar, Alberto 39
Hall, Peter M. 63
Ma, Jixin 73
Martínez de la Escalera, Nieves 13
Martínez-Herrera, A. F. 119
Méndez de la Torre, José C. 39
Mex-Perera, Carlos 131
Michel, EMR 49
Millán, Valeria 13
Monroy, Raul 131
Morales Acoltzi, Tomás 95
Neme, Antonio 27
Neme, Omar 27
Petridis, Miltos 73
Pulido, JRG 49
Reyes, G 49
Sidorov, Grigori 73
Suarez-Hernandez, David 83
Terashima-Marín, Hugo 109
Torres-Román, D. 119
Trueba Vázquez, Leopoldo 39
Valenzuela-Rendón, Manuel 109
Vargas-Rosales, C. 119
Velarde-Alvarado, P. 119
Verduzco-Reyes, Gustavo 3
Villaseñor, Elio 13
Viveros Jiménez, Francisco 95
Xiao, Bai 63
Zamora-Elizondo, José 131
Zhao, Guoxing 73

Editorial Board of the Volume

Impreso en los Talleres Gráficos
de la Dirección de Publicaciones
del Instituto Politécnico Nacional
Tresguerras 27, Centro Histórico, México, D.F.
Octubre de 2008
Printing 500 / Edición 500 ejemplares

This volume contains contributions on selected topics of Artificial Intelligence.

The following areas are represented in this volume:

- Self-organizing Maps,
- Graph Matching and Pattern Recognition,
- Computer Security.

In each area, the contributions of this volume are related with development of algorithms and applications.

The volume will be useful for researchers, students, and general public interested in the corresponding areas of Artificial Intelligence.

ISSN: 1870-4069

www.ipn.mx
www.cic.ipn.mx

INSTITUTO POLITÉCNICO NACIONAL
"La Técnica al Servicio de la Patria"