Recommendation of Process Discovery Algorithms: a Classification Problem

Damián Pérez-Alfonso, Raykenler Yzquierdo-Herrera, and Manuel Lazo-Cortés

University of Informatics Sciences, Havana, Cuba
{dalfonso,ryzquierdo,manuelslc}@uci.cu

Abstract. Process mining techniques extract knowledge from event logs of information systems. Process discovery is a process mining category, focused on discovering process models. The applicability and effectiveness of process discovery algorithms depend on event log's features. Selecting the right algorithms is a tough task due to the variety of variables involved and the complexity of obtaining logs features. To choose a suitable discovery algorithm the traditional approaches use empirical assessment. The metrics to perform this assessment are not applicable to all algorithms. Besides, empirical evaluation is time consuming and computationally expensive. The present paper proposes a new approach that, based on event log characteristics, recommends the discovery algorithms to be used. A new technique of sub-processes diagnosis is proposed for characteristics extraction. The recommendation procedure is formalized as a typical classification problem. This approach could be useful for large event logs and unclear processes analysis.

Key words: process discovery, process mining, classfication

1 Introduction

Information systems record in event logs the execution of supported business processes. Process mining involves discovery, conformance and enhancement of process starting from event logs. Performance evaluations, anomaly identification, compliance checking, among other kinds of analysis, need models that accurately reflect the actual execution of processes. This need has driven to the development of a variety of algorithms for discovering process models.

A process discovery algorithm is a function that maps an event log onto a process model, such that the model is "representative" for the behavior seen in the event log [1]. Noise, duplicate tasks, hidden tasks, non-free choice constructs and loops are typical problems for discovery algorithms [2]. Other problems are related to the mining of unstructured processes, commonly present in real environments [3]. Full or comprehensive solutions to the aforementioned challenges have not been submitted in literature. Thus, algorithms effectiveness depends on event log characteristics and their associated process.

The varying performance of discovery algorithms creates uncertainty during its application. Obtaining a quality model could require the use of several algorithms interchangeably, thus becoming a time consuming task. Selecting the

Special Issue: Advances in Pattern Recognition
Research in Computing Science 61, 2013, pp. 33-42
Paper Received 20-03-2013 and Accepted 22-04-2013

right algorithms is a hard task due to the variety of variables involved and the complexity of obtaining event logs properties.

A set of techniques for evaluating discovery algorithms has already been developed. Two kinds of metrics are used: metrics based on comparing the behavior in the discovered model with the behavior in the log, and metrics based on comparing the discovered model against a reference model related to the process [4]. Several techniques executes different discovery algorithms for an event log and evaluates resulting models using both kinds of metrics. Nevertheless, using this empirical evaluation approach for every event log is computationally expensive and time consuming. An approach that overcomes these shortcomings requires reference models [5]. However, reference models are not commonly available in contexts where process discovery is required. If there are reference models, it is unwise to assume that they reflect the actual execution of processes.

Studies that attempt to establish the algorithms with better performance under certain conditions have been published using the aforementioned empirical evaluation techniques [2]. But recognizing these conditions in real environments is a complex task. Also, the impact of each condition on model quality is not clearly defined yet. The existing metrics are not applicable to all algorithms due to modeling notation issues. Therefore, the actual use of these studies remains limited.

The aim of this paper is to establish the necessity and feasibility of a new approach to select discovery algorithms. In this paper, process discovery challenges are analyzed. Also, a critical review of main evaluation and selection techniques, so far proposed, is carried out. Through a novel technique of sub-processes diagnosis, it is possible to extract event log features such as: control-flow patterns, invisible tasks and infrequent behavior (noise). Therefore, it is feasible to construct a recommendation system of discovery algorithms starting from event logs characteristics. The recommendation procedure for this system is formalized as a typical classification problem.

The paper is structured as follows. Difficulties of process discovery are presented in the next section. Section 3 provides a literature review of techniques and approaches for evaluation and recommendation of discovery algorithms. In Section 4, the proposal of this paper is explained: firstly the factors to consider for a comprehensive mechanism of recommendation are established, usefulness of a new diagnosis technique for extracting log features is described, and finally, an outlook on some aspects for dealing with the recommendation of algorithms as a classification problem is projected. The last section is devoted to conclusions and outlines for future work.

2 Process Discovery Challenges

In order to properly select a discovery algorithm it is important to master the difficulties of process discovery. For a comprehensive and accurate recommendation all these difficulties and their influence on algorithm performance must be taken into account.

Heterogeneity of data sources from real environments, among other reasons, can lead to difficult cases for discovery algorithms [6]. Infrequent traces and data recorded incompletely and/or incorrectly can induce wrong interpretations of process behavior. Moreover, data provided by parallel branches and ad-hoc changed instances generates complex sequences on event logs, creating traces that are harder to mine.

Process structure is another source of challenges for discovery algorithms. Presence of control-flow patterns like non-free choices, loops (nested or not) and parallelism affect the discovery algorithms. For example, algorithms like α, α^+, $\alpha^\#$ and α^* do not support non-free choices [7]. On the other hand, DWS Mining and α^{++} can deal with non-free choice but cannot support loops [2].

The discovery of a process model requires that the event log contains enough information, i.e. has a level of completeness such that its traces are representative of process behavior. Completeness of event logs can be affected by absent information, i.e. existence of invisible tasks in the process. FSM Miner/Genet, β (Tsinghua α), α, α^+ and α^{++} are algorithms affected by invisible tasks.

Obtaining a quality model is another challenging aspect in process discovery. There are various metrics and approaches for estimating process model quality, though there is a consensus on the following quality criteria presented by Aalst [1]:

- *Fitness:* The model should allow the behavior present in the event log.
- *Precision:* The model should not allow a behavior that is completely unrelated to that present in the log.
- *Generalization:* The model should generalize the behavior present in the log.
- *Simplicity:* The model should be as simple as possible. Also referred as structure, is influenced by the vocabulary of modeling language.

These are competing criteria because there is an inverse relationship between generalization and precision. A too general model could lead to allow much more behavior than present in the log, also known as underfitting model. On the contrary, a too precise or overfitting model is undesirable.

The right balance between overfitting and underfitting is called *behavioral appropriateness*. The *structural appropriateness* of a model, on the other hand, refers to its ability to clearly reflect the performance recorded with minimal possible structure [8]. A quality model requires both behavioral appropriateness and structural appropriateness [9]. It can be appreciated that it is difficult to achieve a proper balance between the abovementioned quality criteria.

In real environments it is common to find unstructured processes. Many business processes are not orchestrated by workflow tools [3]. Furthermore, even when processes must be executed according to a designed process model, in practice, some room for flexibility is necessary for smooth operation of enterprise [10]. Both situations create unstructured processes, a challenging process type for discovery algorithms. The large number of alternative flows contained by logs from unstructured process complicates the detection of control-flow patterns and leads to complex and poorly understandable models.

3 Related work

In order to identify which discovery algorithm allows to obtain suitable models for particular situations, a set of techniques for algorithms evaluation have been developed. Performance of these algorithms is determined through evaluation of quality of obtained models. Defined quality metrics are grouped under two main methods [4]. One method compares the discovered model with respect to the event log and is called *model-log*. The other method, called *model-model*, assesses similarity between discovered model and a reference model of process.

Rozinat et al. devised an evaluation framework that allows end users to validate the results of the application of process mining, and researchers in this area to compare the performance of discovery algorithms [11]. The proposal combines the above mentioned evaluation methods.

The framework uses two approaches: assessment of the quality of discovered models through existing evaluation metrics and a *k-fold cross validation*, an evaluation technique from machine learning domain. The negative examples needed for the *k-fold cross validation* are obtained by generating a random event log. This approach for negative examples generation may include false negatives.

Use of this framework as a recommendation mechanism is not suitable owing to the cost involved on empirical assessments of discovery algorithms. The *k-fold cross validation* requires k executions for each algorithm to be assessed. In the other approach several executions are needed by each algorithm too, because of different inputs of evaluation metrics. The metrics used are not directly comparable to each other as they measure different aspects of quality at different levels of granularity and they are defined for different modeling notations [11]. This also limits the use of this framework for recommendation of discovery algorithms.

Following the *model-log* method, Ma proposes another evaluation framework [12]. The main novelty of this framework is the inclusion of a parameter optimization step using *k-fold cross validation.* The parameters optimization is a key issue since its values affect the obtained model and thus the performance of discovery algorithms [13]. Negative examples are used in this framework for the evaluation stage and parameters optimization.

For the Ma framework, the empirical evaluations cost is also the main shortcoming for its use as a recommendation tool. In the experiments with complex event logs, the negative examples generation through AGNES [14] created serious performance problems [12]. This proposal has also limitations related to the modeling formalism supported by the selected metrics, so it is not applicable to any discovery algorithm.

De Weerdt et al complement the *model-log* evaluation method with a comprehensibility assessment of the models obtained [2]. Comprehensibility is associated to the quality criteria called simplicity. In [2] the performance of seven discovery algorithms was analyzed on real and artificial logs. Using statistical techniques was concluded that complexity of event log has an important impact on the evaluated quality criteria. Also, were found remarkable differences between assessment on artificial logs and assessment on real logs.

Some relationships found among simple event log features and the results of the discovery algorithm were highlighted in these paper. Nevertheless, these relationships must be quantified in order to be used for algorithm recommendation. It is also required to include the impact of more complex features such as noise, lack of information and completeness.

All evaluation techniques analyzed so far, apply mainly to algorithms that can discover models on Petri nets or are transformable to Petri nets, due to the preponderance of metrics associated with this notation. This limitation excludes algorithms such as Fuzzy Miner [15] and techniques like pattern abstractions or sequence clustering, which are particularly useful for discovering unstructured processes, a context where algorithms that generate Petri nets are ineffective [2]. Additionally, in real environments where process models are not specified or do not reflect the actual processes execution, the *model-model* evaluation method is not applicable. Moreover, selecting a discovery algorithm for a given situation based on empirical evaluation, involves time and resource consumption for each of the algorithms chosen as a possible solution.

3.1 Beyond empirical evaluation

Wang et al.'s [5] work is a major effort to minimize the empirical evaluation problem in selection of discovery algorithms. The proposed framework is based on selecting reference models of high quality and building from these a regression model to estimate the similarity of other process models. This approach is comprised of a learning phase and a recommendation phase.

With the structural characteristics of the significant references models and the similarity values obtained after empirical evaluation of algorithms, a regression model is constructed during the learning phase. In the recommendation phase the reference models features are extracted in order to predict the similarity results using the regression model. Starting from the estimated similarity values the ideal algorithm for discovering the processes associated to the reference model is proposed.

The experiments conducted using this recommendation solution show encouraging results [5]. The evaluation over a set of 621 models (including artificial and real models) achieved more than 90 % accuracy in the recommendations. However, this approach involves some requirements that severely limit its application.

The main constraining requirement is the need of reference models for the evaluation and prediction. In multiple real-world environments, where discovery algorithms need to be applied, the process models are not described or are inconsistent and/or incomplete. Weng et al.'s approach assumes that the actual execution of the processes keeps a close relationship with their reference model. Therefore, in contexts where features of the actual logs differ from logs artificially generated by the reference models inexact results can be expected. The construction of regression model from model features discards issues like noise, lack of information and completeness of event logs, which have a significant impact on the performance of discovery algorithms.

Last but not least, the proposed framework has been conceived for using labeled Petri nets as the modeling language. Therefore, it is only applicable in environments where the models are specified using Petri nets or equivalent notations. This constraint also implies that the framework could only recommend algorithms based on Petri net or equivalent notations, excluding algorithms such as the Fuzzy Miner. As mentioned before, this algorithm is especially useful in contexts where discovery algorithms that generate Petri nets are inefficient.

An alternative direction to solve the algorithm selection problem is offered by Lakshmanan and Khalaf [16]. The authors construct a decision tree starting from the comparison of five discovery algorithms. The comparison establishes the algorithms potentialities to tackle challenges like: invisible and duplicate tasks, loops, parallelisms, non-free choice and noise.

Nevertheless, it is not specified how to identify the presence of challenging situations in the process to mine. While recognition of complicated control-flow patterns can be performed from a reference model, this implies drawbacks already discussed. Moreover, identifying invisible tasks, duplicate tasks, noise, loops, parallelisms or non-free choice from an event log is far from trivial. In general, this study provides some important theoretical elements but lacks direct practice applicability due to the required information type.

4 Proposal

Taking into account the identified challenges for discovery algorithms and the aforementioned limitations of existing solutions for selection and recommendation, it can be established that the conception of a comprehensive mechanism for recommendation of discovery algorithms should consider the following factors:

1. The event log is the main information source that is available in all environments for process characterization.
2. The peculiarities of event log such as noise, lack of information, completeness and log size (number of event classes, number of traces, etc.) must be considered in addition to process characteristics.
3. The process characteristics and the event log peculiarities are not always fully identifiable.
4. There should be no limitations regarding modeling notation of the discovery algorithms to recommend.
5. The process discovery goal may require to reinforce some of the desired quality criteria in the model to be obtained.

Considering factors 1 and 2 there is a need for techniques capable of extracting information from an event log that could be useful for the recommendation. A new technique has been developed to obtain relative values of event log completeness starting from certain basic assumptions [17]. This technique employs statistical estimators and is implemented as a ProM plug-in. A new diagnostic tool based on sub-processes decomposition is specially useful to identify other event log peculiarities and process characteristics.

4.1 Sub-process decomposition

A new technique for estimating the lack of information in the event logs has been developed [18]. This technique is able to recognize the control-flow patterns present in an event log starting from its trace alignment [19]. The identified control-flow patterns allow to decompose the process into sub-processes. The decomposition is represented by a tree named tree of building blocks.

The sub-processes are the leaves in the tree of building blocks. Each leaf shows the trace alignment of the sub-processes and the frequency of the different cases. This information facilitates the identification of cases that potentially represent noise in each of the sub-processes. Based on the tree of building blocks this tool identifies patterns of lack of information and estimates the associated invisible tasks. As can be seen above, this tool identifies, from an event log, a set of characteristics that affect the performance of the discovery algorithms.

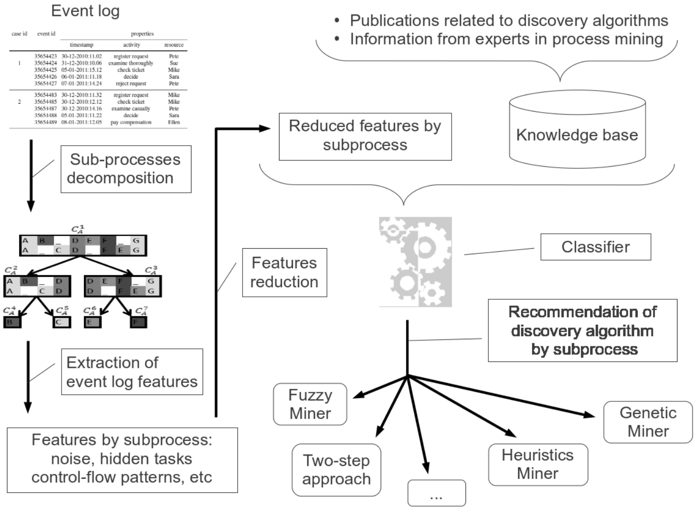
It can also be particularly useful to segment large and/or complex event logs in a meaningful way. The sub-processes generally possess different features: presence of different control-flow patterns, different types of trace frequency, existence of patterns of lack information. These features determine the effectiveness of discovery algorithms, as it has already been explained. Therefore, it is suggested to perform the recommendation for each one of the sub-processes because even if they belong to the same event log, different sub-processes may require different discovery algorithms.

4.2 Recommendation as a classification problem

Classification is the problem concerning the construction of a procedure that will be applied to a continuing sequence of cases, in which each *new case* must be assigned to one of a set of *pre-defined classes* on the basis of *observed attributes or features* [20]. The recommendation of discovery algorithms can be expressed in terms of a classification problem. An event log on which is necessary to recommend a discovery algorithm is considered as a *new case* to be classified. The recommended algorithm is the *pre-defined class* to be assigned to an event log based on its *observed features*. This kind of classification procedure where the true classes are known has also been variously termed as pattern recognition, discrimination, or supervised learning [20].

Typifying the recommendation of discovery algorithms as a classification problem opens up the way to the application of techniques developed in a long-standing knowledge area. Techniques from the *classification* area have already been used for the recommendation of discovery algorithms [5]. Nevertheless, for the conception of a comprehensive recommendation mechanism that overcomes the limitations of existing solutions, it is important to incorporate to this classification problem the aforementioned factors. The recommendation mechanism proposed can be observed in Fig. 1.

According to the first of the aforementioned factors, the features for the classification of the *new case* should be extracted just from the event log. It should be noted that the characteristics mentioned in the second factor differ in



case id	event id	properties		
		timestamp	activity	resource
1	35654423	30-12-2010:11.02	register request	Pete
	35654424	31-12-2010:10.06	examine thoroughly	Sue
	35654425	05-01-2011:15.12	check ticket	Mike
	35654426	06-01-2011:11.18	decide	Sara
	35654427	07-01-2011:14.24	reject request	Pete
2	35654483	30-12-2010:11.32	register request	Mike
	35654485	30-12-2010:12.12	check ticket	Mike
	35654487	30-12-2010:14.16	examine casually	Pete
	35654488	05-01-2011:11.22	decide	Sara
	35654489	08-01-2011:12.05	pay compensation	Ellen

Event log
Sub-processes decomposition
C_A^1
A B _ D E F _ G
A _ C D _ F E G
C_A^2
A B _ D
A _ C D
C_A^3
D E F _ G
D _ F E G
C_A^4
B
C_A^5
C
C_A^6
E
C_A^7
F
Extraction of event log features
Features by subprocess: noise, hidden tasks control-flow patterns, etc
Features reduction
Reduced features by subprocess
• Publications related to discovery algorithms
• Information from experts in process mining
Knowledge base
Classifier
Recommendation of discovery algorithm by subprocess
Fuzzy Miner
Two-step approach
...
Heuristics Miner
Genetic Miner

Fig. 1. Recommendation of discovery algorithms through sub-processes decomposition.

their values scales because in this classification problem the *cases features* have different orders of magnitude.

To solve a classification problem the design of the classifier is an essential issue. Roughly speaking, there are three different approaches for designing a classifier. The first approach is based on the concept of similarity, the second one is a probabilistic approach and the third approach is to construct decision boundaries directly by optimizing certain error criterion [21]. Selecting the right approach and designing an efficient classifier is a complex task that exceeds the scope of this paper. However, taking into account factor 3, it is useful to stress that the classifier which will be designed for the recommendation of algorithms should be able to deal with incomplete information about *cases features.*

A major challenge for this classification problem is the building of a knowledge base. Useful information for the knowledge base can be provided by a set of publications and experts in the field of process mining. Published results about assessment of discovery algorithms using traditional approaches are a good starting point. Nevertheless, results tied to existing quality metrics do not cover all the algorithms due to modeling notations issues. Considering the factor 4, information about performance of algorithms that cannot be assessed using quality metrics is needed. Therefore, the building of the knowledge base for algorithms recommendation opens up the opportunity for further research.

Several alternatives can be assessed to incorporate the factor 5 into the classification problem. To include in the classes definition the combination of the discovery algorithm and the quality criterion to reinforce is an alternative. To include the quality criterion to reinforce as part of the characterization of cases to classify could be another alternative.

Performing the recommendation of discovery algorithms through classification requires further research. The existing knowledge in the process mining area should be recovered and structured it on one of the existing representations for that purpose. An efficient classifier from existing approaches must be designed. The feature selection procedure to apply is needed. Nevertheless, in order to formalize the classification solution some insights were provided taking into account the aforementioned factors for the conception of a comprehensive recommendation mechanism.

5 Conclusions and Future Work

Characteristics such as noise, duplicate tasks, hidden tasks, non-free choice constructs and loops could affect the performance of discovery algorithms. Current approaches that select discovery algorithms based on empirical assessments are computationally expensive and time consuming. Other approaches presuppose the existence of reference models; however, there is a low probability of having available reference models. In general, the approaches based on assessment have shortcomings that are induced by the existing metrics. Besides, the shortcomings are related to the modeling notation of the discovery algorithms.

This paper presented a new approach that based on event log characteristics recommends the discovery algorithms to be used. Important factors to be considered in the conception of a comprehensive mechanism to recommend discovery algorithms were declared in the proposal. The technique for sub-processes diagnostic described in Section 4.1 would be especially useful for recommendation based on event logs characteristics.

The recommendation of discovery algorithms can be treated as a typical classification problem. Basic ideas to formalize that classification problem were presented. To solve the recommendation of discovery algorithms as a classification problem further research is required in order to build up a knowledge base, to select its most suitable structure, to apply an effective features selection and to design the classifier. A great starting point for those further research works are the available techniques in a long-standing knowledge area like classification.

The recommendation of discovery algorithms using subprocesses and based on event logs characteristics could be useful for process mining projects in enterprises with large and complex event logs.

References

1. van der Aalst, W.M.P.: Process Mining. Discovery, Conformance and Enhancement of Business Processes. Springer, Heidelberg, Dordrecht, London et. al (2011)
2. De Weerdt, J., De Backer, M., Vanthienen, J., Baesens, B.: A multi- dimensional quality assessment of state-of-the- art process discovery algorithms using real- life event logs. Information Systems **37** (March 2012) 654–676
3. Desai, N., Bhamidipaty, A., Sharma, B., Varshneya, V.K., Vasa, M., Nagar, S.: Process trace identification from unstructured execution logs. In: Services Computing (SCC), 2010 IEEE International Conference on. (2010) 17–24

4. De Weerdt, J., De Backer, M., Vanthienen, J., Baesens, B.: A critical evaluation study of model-log metrics in process discovery. Volume 66 LNBIP of 8th International Workshops and Education Track on Business Process Management, BPM 2010., Hoboken, NJ (2011)
5. Wang, J., Wong, R.K., Ding, J., Guo, Q., Wen, L.: Efficient selection of process mining algorithms. IEEE Transactions on Services Computing **99**(1) (2012) 1–1
6. Ly, L.T., Indiono, C., Mangler, J., Rinderle-Ma, S.: Data transformation and semantic log purging for process mining. In: 24th International Conference on Advanced Information Systems Engineering (CAiSE'12). LNCS, Springer (2012)
7. Van Dongen, B., Alves de Medeiros, A., Wen, L.: Process mining: Overview and outlook of petri net discovery algorithms. In: Transactions on Petri Nets and Other Models of Concurrency II. Volume 5460 of Lecture Notes in Computer Science., Springer (2009) 225–242
8. van der Aalst, W.M.P., Rubin, V., Verbeek, H., Van Dongen, B., Kindler, E., Günther, C.: Process mining: A two-step approach to balance between underfitting and overfitting. Software and Systems Modeling **9**(1) (2010) 87–111
9. Rozinat, A., van der Aalst, W.M.P.: Conformance testing: Measuring the fit and appropriateness of event logs and process models. In: Third International Conference on Business Process Management(BPM 2005), France (2006) 163–176
10. Mieke Jans, Michael Alles, Miklos Vasarhelyi: Process mining of event logs in internal auditing: A case study. (2011)
11. Rozinat, A., Medeiros, A.K.A.d., Günther, C.W., Weijters, A.J.M.M., van der Aalst, W.M.P.: Towards an evaluation framework for process mining algorithms. BPM Center Report (2007)
12. Ma, L.: How to Evaluate the Performance of Process Discovery Algorithms. Master thesis, Eindhoven University of Technology, Netherlands (2012)
13. Weijters, A.: An optimization framework for process discovery algorithms. In Stahlbock, R., ed.: Proceedings of the International Conference on Data Mining, Las Vegas Nevada, USA (2011)
14. Goedertier, S., Martens, D., Vanthienen, J., Baesens, B.: Robust process discovery with artificial negative events. Journal of Machine Learning Research **10** (2009)
15. Günther, C., van der Aalst, W.M.P.: Fuzzy mining - adaptive process simplification based on multi-perspective metrics. In: Business Process Management. Volume 4714 LNCS of 5th International Conference on Business Process Management, BPM 2007., Brisbane (2007) 328–343
16. Lakshmanan, G., Khalaf, R.: Leveraging process mining techniques to analyze semi-structured processes. IT Professional **99**(PrePrints) (2012) 1–1
17. Yang, H., van Dongen, B.F., ter Hofstede, A.H.M., Wynn, M.T., Wang, J.: Estimating completeness of event logs. (2012)
18. Yzquierdo-Herrera, R., Silverio-Castro, R., Lazo-Cortés, M.: Sub-process discovery: Opportunities for process diagnostics. In Poels, G., ed.: Enterprise Information Systems of the Future. Number 139 in Lecture Notes in Business Information Processing. Springer Berlin Heidelberg (2013) 48–57
19. Bose, R.P.J.C., van der Aalst, W.M.P.: Process diagnostics using trace alignment: Opportunities, issues, and challenges. Information Systems **37**(2) (April 2012) 117–141
20. Michie, D., Spiegelhalter, D.J., Taylor, C.C., Campbell, J., eds.: Machine learning, neural and statistical classification. Ellis Horwood, Upper Saddle River, NJ, USA (1994)
21. Zheng, L., He, X.: Classification techniques in pattern recognition. In: WSCG, conference proceedings, ISBN. (2007) 80–903100