



HAL
open science

Extraction of Agent Groups with Similar Behaviour Based on Agent Profiles

Kateřina Slaninová, Jan Martinovič, Roman Šperka, Pavla Dráždilová

► **To cite this version:**

Kateřina Slaninová, Jan Martinovič, Roman Šperka, Pavla Dráždilová. Extraction of Agent Groups with Similar Behaviour Based on Agent Profiles. 12th International Conference on Information Systems and Industrial Management (CISIM), Sep 2013, Krakow, Poland. pp.348-357, 10.1007/978-3-642-40925-7_32 . hal-01496081

HAL Id: hal-01496081

<https://inria.hal.science/hal-01496081v1>

Submitted on 27 Mar 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Extraction of Agent Groups with Similar Behaviour Based on Agent Profiles

Kateřina Slaninová¹, Jan Martinovič², Roman Šperka¹, and Pavla Dráždilová²

¹ School of Business Administration in Karviná,

Silesian University in Opava,

Univerzitní nám. 1934/3a, 733 40, Karviná, Czech Republic

slaninova@opf.slu.cz, sperka@opf.slu.cz

² Faculty of Electrical Engineering,

VŠB Technical University in Ostrava,

17. listopadu 15/2172, 708 33 Ostrava, Czech Republic

jan.martinovic@vsb.cz, pavla.drazdilova@vsb.cz

Abstract. This paper deals with the log files suitable to extract valuable information about agents and their behaviour from agent-based simulation in a model of virtual company. Such information, presented in a transparent way, can be used as a support for simulation verification to achieve the suitable design of the proposed system. Hence, based on the similar behaviour (represented by extracted sequences) of agents, we are able to construct models which explain certain aspects of agent behaviour. Moreover, we can extract agent profiles based on behaviour and find latent ties between different agent groups with similar behaviours. The paper extends the results of our previous works about sequence extraction and comparison. The approach for agent network construction based on agent profiles is described. Two different methods were used for construction of agent network. One method uses cosine similarity and graph partitioning and the second self organization maps and Euclidean similarity for agent relations. Each of these methods has its advantages and disadvantages which are summarized in the paper and presented in the form of the visualization of relations between agents.

Keywords: Agent Profile, Log Analysis, Agent Behaviour

Introduction

Modern applications like information systems, enterprise systems or e-commerce systems, as well as monitoring applications, web applications, and other systems produce a large amount of data collections. These data collections are usually kept in databases, data warehouses, or simply in data or log files.

A standard log file typically consists of records with information about recorded events that have occurred in the system. These records may contain various attributes such as information about the date and time when the event happened, an originator, the type of event and additional information. The originator can be a person, a device, or software. This depends on the type of log file. In the case of a person, we can extract the relevant records and obtain information about his behaviour.

This paper deals with the log files suitable to extract valuable information about agents and their behaviour from agent-based simulation in a model of virtual company [1]. The motivation to use agents as simulation subjects outcomes from the agents characteristics e.g. autonomy, coordination, communication etc. precisely declared in [2, 3]. This ensures to substitute real business processes participants (sellers and customers) and simulate their trading communication. Such information, presented in a transparent way, can be used as support for simulation verification to achieve the suitable design of the proposed system.

This paper is focused on log files where one log file attribute is an originator of the recorded activity, and the originator is an agent. Hence, based on the similar attributes of agents, we are able to construct models which explain certain aspects of an agent behaviour [4]. Moreover, we can extract agent profiles based on behaviour and find latent ties between different agent groups with similar behaviours.

1 Log Analysis

Log analysis is a data mining process focused on the analysis of computer-generated records (also called audit trail records, event logs or transaction logs) [5]. Log file analysis has gained growing attention in all areas of human activity. This domain is very interesting not only for researchers, but for commercial software developers as well. There are various disciplines considered in the analysing of data sources with the intention of achieving worthy information, often represented as knowledge. Obtained information (or knowledge) is then used for management, system maintenance, and system optimisation, marketing campaigns, recommender systems or other purposes such as discovering a company's structure or the structure of social networks.

A log file is usually a simple text file generated by a device, software, application or system. It consists of messages, which are represented by records of events performed in the system. An event log consists of cases, whilst cases consist of events. It typically contains activity information in events, but mostly timestamp, originator (performer) and other necessary data as well.

In a case, the events are represented in the form of a *trace*, i.e. a sequence of unique events [4, 6]. If an event log contains timestamps (a time when the event was performed in the system) then the ordering in a trace respects this attribute $\#_{time}(e)$.

2 Pattern Mining

Unsupervised pattern mining is used for pattern recognition tasks, where training feature vectors with known class labels are not available. In this type of problem, there is used a set of feature vectors \mathbf{x} , and the main goal is to achieve clusters (groups) of vectors on the basis of their similarity (or proximity measure). This process is generally known as clustering.

In the experiments, selected pattern mining methods were used to find patterns of similar behaviour of agents and to find groups of agents with similar behaviour. This section is only short introduction of the selected methods, used in the case study.

2.1 Self Organizing Maps

Self organizing maps (SOM), also called Kohonen maps, is a type of artificial neural network invented by professor Teuvo Kohonen in 1982 [7]. The input space of the training samples is represented in a low dimensional (often two-dimensional) space, called map. The model is capable of projecting a high-dimensional space to a lower-dimensional space and is efficient in structure visualization due to its feature of topological preservation using a neighbourhood function. Obtained low-dimensional map is often used for pattern detection, clustering, or for characterization and analysis of the input space [8]. The Euclidean distance is used in the SOM algorithm.

2.2 Spectral Clustering

Spectral clustering is one of the divisive clustering algorithms which can be applied in the graph theory. The spectral clustering algorithm uses eigenvalues and eigenvectors of a similarity matrix derived from the data set to find the clusters. In this section, there is described the type of spectral clustering based on the second smallest eigenvector of the Laplacian matrix.

How to use the spectral algorithm is studied in [9] by Cheng et al. In [10] Ding et al. proposed a new graph partition method based on the min-max clustering principle: the similarity between two subgraphs (cut set) is minimized, while the similarity within each subgraph (summation of similarity between all pairs of nodes within a subgraph) is maximized.

Spectral clustering method optimized by our proposed Left-Right Oscillate algorithm [11] was used in two parts of the proposed approach: for finding the behavioural patterns and for better visualization of agent network based on their similar behaviour in the system.

3 Business processed and Agent-Based Simulation

The main motivation for the research in the field of business process simulation was to investigate agent' behavioural patterns and to find groups of agents with similar behaviour during the agent-based simulation in a model of virtual company. Used multi-agent system (MAS) operates with randomly (resp. pseudo randomly) generated parameters, and is able to deal with unpredictable phenomena surrounding the company. To achieve the suitable design of the proposed system, a mechanism for verification is needed.

Agent-based simulations dealing with a company simulation can bring several crucial advantages [12, 13]. It is possible to involve unpredictable disturbance of the environment into the simulation with the agents. The analysis and visualization of agent behaviour lead to facilitate the verification of the simulation model and to effectively observe the reaction of the model in relation to the initialization of its input attributes, or in relation to the influence of the model environment.

3.1 Simulation Model

The agent-based simulation model from which were analysed data outputs is described in [14], with the focus on Business Process Management (BPM). Usual business process simulation approaches are based on the statistical calculation, as can be seen for example in [15]. However, only several problems can be identified while using the statistical methods. The model uses the advantages of the agent technology, to describe some of the core business processes of a typical business company. The authors in [16] developed a business process simulation framework in order to simulate the real behaviour of the company on the market.

The core of the experiments was the analysis of the agent-based simulation outputs. To ensure the outputs, above mentioned framework was used to trigger simulation experiments. The simulation framework covered business processes supporting the selling of goods by company sales representatives to the customers.

Each activity performed by agents in the simulation model was recorded into a log file. The log was a simple text file, where each row represented one event. The main row structure, which was valid for all agents, consisted of following event attributes: $\#_{time}(e)$, denoted as `TimeStamp` (date and time when the event was performed), $\#_{resource}(e)$, denoted as `AgentID` (each agent had its unique ID), $\#_{activity}(e)$, denoted as `TypeOfAction`, and other additional attributes like `AgentClassName` or `ActionAttribute`. Example of the log file is presented in Example 1.

Example 1 (Example of Log File - Simulation Model).

```
2012/03/26 01:14:14 : SA0022 bpm.selling.SellerAgent SENT_PROPOSAL CA0223,12,5.0006104,73
2012/03/26 01:14:14 : SA0008 bpm.selling.SellerAgent SENT_PROPOSAL CA0088,8,5.0097656,31
...
2012/03/26 01:14:15 : CA0183 bpm.selling.CustomerAgent RECEIVED_CPF_MESSAGE SA0018,8,5.004883,53
2012/03/26 01:14:15 : CA0170 bpm.selling.CustomerAgent SENT_REPLY SA0017,false,1.0462701894732744,5.0006104,29
2012/03/26 01:14:15 : CA0170 bpm.selling.CustomerAgent FINISHED_TURN
```

In used multi-agent model, there were monitored five types of agents: `SellerAgent`, `CustomerAgent`, `ManagerAgent`, `InformativeAgent` and `DisturbanceAgent`.

4 Proposed Approach with Case Study in Business Process Simulation

The proposed approach proceeds from the original social network approach with a modification focused on agent behaviour. The modification is based on a definition of the relation between agents. The original approach to the analysis of social networks deals with the assumption that the social network is a set of people (or groups of people) with social interactions among themselves [17]. Social interaction is commonly defined as interaction between the actors like communication, personal knowledge of each other, friendship, membership, etc...

The modification extends the original approach of social network analysis by the perspective of the complex networks. This type of view differs from the original approach in the description of relations between the nodes (agents). The relation between the agents is defined by their common attributes characterizing their behaviour in the system. More specifically, the agent behaviour in the system is defined by agent profiles.

Creation of the agent profiles requires a set of agents U and a set of event sequences S performed in the system by the agents. Extraction of these sets is described in our article [18].

Definition 1. (Base agent profile, sequences)

Let $U = \{u_1, u_2, \dots, u_n\}$, be a set of agents, where $i = 1, 2, \dots, n$ is a number of agents u . Then, sequences of events $\sigma_{ij} = \langle e_{ij1}, e_{ij2}, \dots, e_{ijm_j} \rangle$, are sequences of events executed by the agent u_i in the system, where $j = 1, 2, \dots, p_i$ is a number of that sequences, and m_j is a length of j -th sequence. Thus, a set $S_i = \{\sigma_{i1}, \sigma_{i2}, \dots, \sigma_{ip_i}\}$ is a set of all sequences executed by the agent u_i in the system, and p_i is a number of that sequences.

Sequences σ_{ij} extracted with relation to certain agent u_i are mapped to set of sequences $\sigma_l \in S$ without this relation to agents: $\sigma_{ij} = \langle e_{ij1}, e_{ij2}, \dots, e_{ijm_j} \rangle \rightarrow \sigma_l = \langle e_1, e_2, \dots, e_{m_l} \rangle$, where $e_{ij1} = e_1, e_{ij2} = e_2, \dots, e_{ijm_j} = e_{m_l}$.

Define matrix $B \in N^{|U| \times |S|}$ where

$$B_{ij} = \begin{cases} \text{frequency of sequence } \sigma_j \in S \text{ for agent } u_i & \text{if } \sigma_j \in S_i \\ 0 & \text{else} \end{cases}$$

A base agent profile of the agent $u_i \in U$ is a vector $b_i \in N^{|S|}$ represented by row i from matrix B .

Example 2 shows an example of base agent profile generated for agent behaviour during a business process simulation in the multi-agent system.

Example 2 (Example of Base Agent Profiles).

```
#agent:sa0028
0|1|ability_generated;|0;
1|253|sent_proposal;sent_proposal;sent_proposal;sent_proposal; ...
3|1|sent_proposal;sold;|6;11;
```

Similarity between two profiles can be quantified by several methods. The selection of the method depends on the type of data used [19]. For Boolean data, Hamming distance, Jaccard dissimilarity or Rogers-Tanimoto dissimilarity may be used. In the case of numerical data, Euclidean distance, Manhattan distance, cosine similarity or correlation distance may be used. If we use string data, then Edit distance, Levenshtein distance or Hamming distance are all potential options. In this paper, the similarity between agent profiles was determined by using Euclidean similarity [20] and cosine similarity.

4.1 Network of Agent Groups with Similar Behaviour

The network of agent groups with similar behaviour was created using graph theory approach. In this phase of the presented approach, it was constructed a weighted and undirected graph $G(V, E)$, where vertices (graph nodes) V were agents u_i and relations (edges) E expressed the similarity between the agent profiles.

The output graph can be obtained by comparison of Base agent profiles. The edge weights were counted by cosine similarity in this paper. As the output, we obtained a graph of agents (a *.gdf* file). Another output was a file with set of agents U and their vectors consisted of sequences (behavioural patterns) and their frequency.

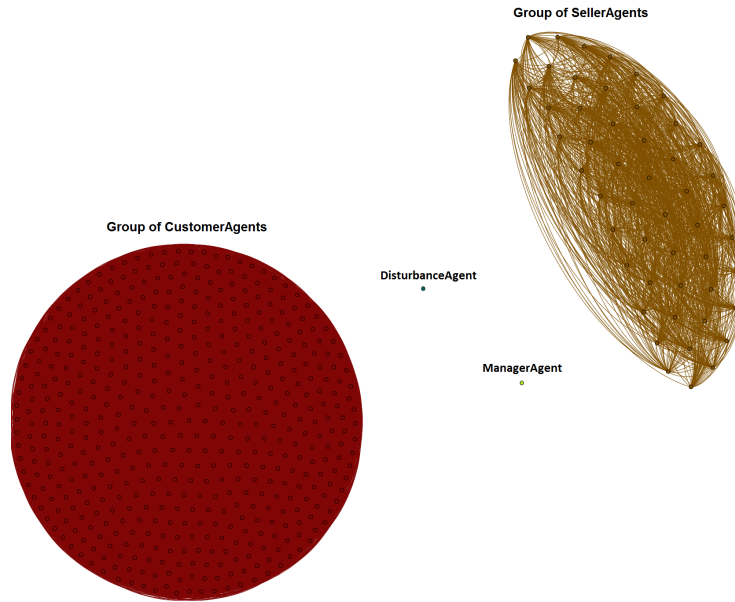


Fig. 1. Example of Agent Network (Spectral Clustering and LRO)

In this last phase of the proposed approach, two methods for construction of network of groups with agents were compared. The first method proceeded from graph of agents, mentioned before. The groups of agents with similar behaviour (similar agent profiles) were found using spectral clustering by Fiedler vector using Left-Right Oscillate algorithm. The graph was constructed only from edges, which weight was higher than a selected threshold. The graph contained the isolated agents (nodes) as well, because they are as important as the other connected nodes. They created a separate component in the graph, see example graph of groups of agents with similar behaviour in multi-agent system in Figure 1.

The second method used for the construction of agent network with groups of agents with similar behaviour was SOM. This method used a file with agents and their vectors consisted of sequences (behavioural patterns) and their frequency as the input for SOM.

The goal of this method was to obtain such output SOM map, where each neuron of the output map consisted of agents with similar behaviour (with similar agent profiles). However, the output SOM map has its predefined structure (in our case lattice), which is not suitable for visualisation of agent network. In Figure 2, we can see example

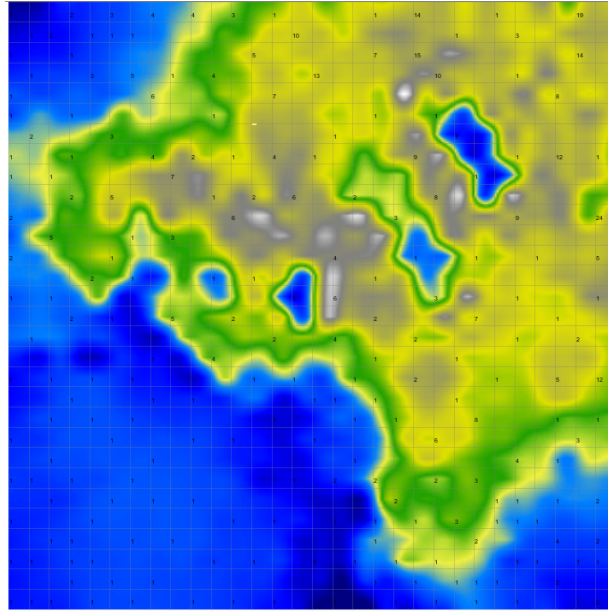


Fig. 2. Example of Clusters with Agents Visualized by SOM

of smoothed data histogram used for cluster visualization in SOM map by SOMToolbox [21]. The example shows the visualization of agents with similar behaviour, where white colour represents neurons with the higher amount of agents and blue colour represents empty (almost empty) neurons.

Therefore, we have used further steps to transform SOM output lattice to graph of groups with agents. This transformation was performed by removing the empty neurons (neurons to which none of agents were assigned). The rest of nodes from output SOM lattice were considered as graph nodes with groups of agents. Due to information about the distance between the neurons, we obtained the weights for relations between the nodes (Euclidean similarity). It is common that some neurons may be very close, which means that the agents in such neurons may be very similar, and may create one group in the agent network. Therefore, a further clustering and colouring of the obtained clusters in a graph of agents was performed using spectral clustering method by Fiedler vector and Left-Right Oscillate algorithm. The different colours represent clusters of agents with similar behaviour.

Moreover, the graph of agents can be reduced into a graph of agent clusters, where clusters contain agents with similar profiles. This approach is suitable for better visualization in cases with large amounts of agents, see example in Figure 4.

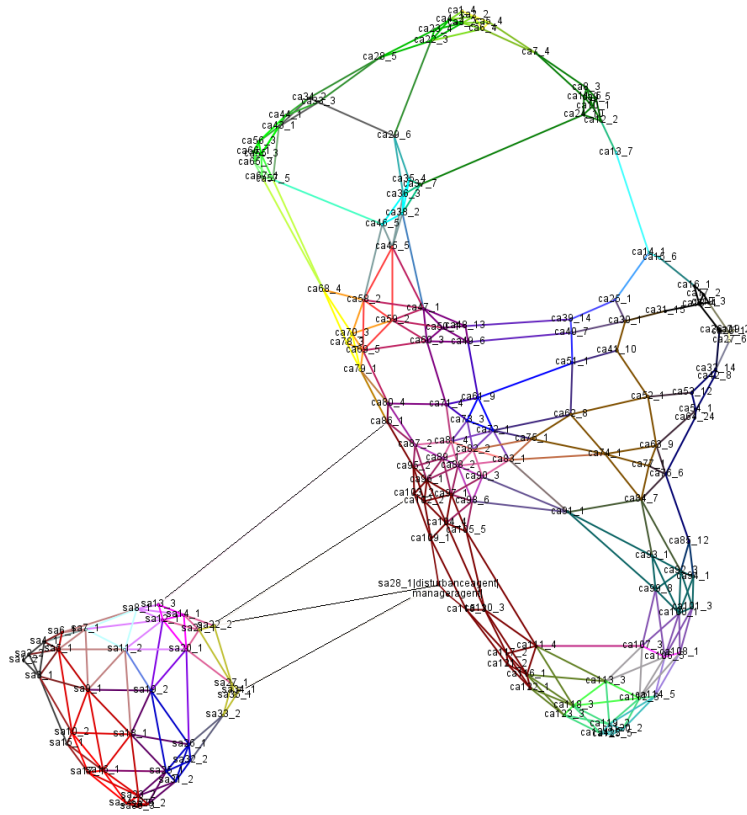


Fig. 3. Example of Agent Network (SOM)

5 Conclusion and Future Work

In the paper, the approach which uses agent profiles for finding of agents groups with similar behaviour have been introduced. This approach may use different methods for groups construction. As founded during presented experiments, each method has its advantages and disadvantages. In the last section, there was presented cosine similarity and graph partitioning method as one possible way for this purpose. Another presented way was based on SOM and graph extraction from SOM output which uses Euclidean similarity.

The results showed, that if we use cosine similarity, the agent without any identical sequences will be in different clusters (groups of agents). The Figure 2 presented results of the tested simulation where two big and compact clusters were found. The problem of this method was, that it was not able to detect tiny changes in the frequency of sequences



Fig. 4. Example of Reduced Agent Network (SOM)

in agent profiles. Therefore, the agent clustering based on their similar behaviour was too rough.

To remove this disadvantage, SOM method was used, and the SOM output map was transformed to the graph. Such approach resolved the problem with detection of tiny changes between agents behaviour. However, this representation had a problem that the relation between totally different profiles could be constructed. This was caused by the reduction to lower dimensions represented by SOM map.

The both disadvantages described above could be resolved by combination of the obtained results. We intent to work on splitting the agents without common sequences in the graph obtained from SOM map. For this purpose, we will use clusters obtained by cosine similarity and graph partitioning.

Acknowledgments. This work was partially supported by the grant of Silesian University in Opava, Czech Republic, No. SGS/06/2013 'Advanced Modeling and Simulation of Economic Systems' and by SGS, VSB – Technical University of Ostrava, Czech Republic, under the grant No. SP2013/167 'Analysis of the behaviour patterns in complex networks'.

References

1. Barnett, M.W.: Modeling & simulation in business process management, Gensym Corporation (2003)
2. Wooldridge, M.: An Introduction to MultiAgent Systems. 2nd edn. Wiley Publishing (2009)
3. Macal, C.M., North, M.J.: Tutorial on agent-based modeling and simulation. In: Proceedings of the 37th conference on Winter simulation. WSC '05, Winter Simulation Conference (2005) 2–15
4. van der Aalst, W.M.P., Reijers, H.A., Song, M.: Discovering social networks from event logs. *Comput. Supported Coop. Work* **14**(6) (2005) 549–593
5. van der Aalst, W.M.P., van Dongen, B.F., Herbst, J., Maruster, L., Schimm, G., Weijters, A.J.M.M.: Workflow mining: a survey of issues and approaches. *Data Knowl. Eng.* **47**(2) (2003) 237–267
6. van der Aalst, W.M.P.: Process Mining: Discovery, Conformance and Enhancement of Business Processes. 1st edn. Springer Heidelberg (2011)
7. Kohonen, T.: Self-Organization and Associative Memory. Volume 8 of Springer Series in Information Sciences. Springer, Berlin, Heidelberg (1984) 3rd ed. 1989.
8. Vojáček, L., Martinovič, J., Slaninová, K., Dráždilová, P., Dvorský, J.: Combined method for effective clustering based on parallel som and spectral clustering. In Snášel, V., Pokorný, J., Richta, K., eds.: DATESO 2011, VŠB - TU Ostrava (2011) 120–131
9. Kannan, R., Vempala, S., Vetta, A.: On clusterings: Good, bad and spectral. *J. ACM* **51**(3) (May 2004) 497–515
10. Ding, C.H.Q., He, X., Zha, H., Gu, M., Simon, H.D.: A min-max cut algorithm for graph partitioning and data clustering. In: ICDM '01: Proceedings of the 2001 IEEE International Conference on Data Mining, Washington, DC, USA, IEEE Computer Society (2001) 107–114
11. Dráždilová, P., Martinovič, J., Slaninová, K.: Spectral clustering: Left-right-oscillate algorithm for detecting communities. In: ADBIS Workshops. (2012) 285–294
12. De Snoo, D.: Modeling planning processes with talmod. Master's thesis, University of Groningen (2005)
13. Jennings, N., Faratin, P., Norman, T., O'Brien, P., Odgers, B.: Autonomous agents for business process management. *Int. Journal of Applied Artificial Intelligence* **14** (2000) 145–189
14. Macal, C., North, J.: Tutorial on agent-based modeling and simulation. In: Proceedings: 2005 Winter Simulation Conference. (2005)
15. Scheer, A., Nuttgens, M.: Aris architecture and reference models for business process management. *Business Process Management, LNCS* **1806** (2000) 376–389
16. Spišák, M., Šperka Roman: Financial market simulation based on intelligent agents - case study. *Journal of Applied Economic Sciences* **VI**(17) (2011) 249–256
17. Newman, M.E.J.: Networks: An Introduction. Oxford University Press (2010)
18. Slaninová, K., Martinovič, J., Dráždilová, P., Vymětal, D., Šperka, R.: Analysis of agents' behavior in multiagent system. In: 24th European Modeling and Simulation Symposium, EMSS 2012. (2012) 169–175
19. Deza, M.M., Deza, E.: Dictionary of Distances. Elsevier Science, Amsterdam, The Netherlands (October 2006)
20. Elmore, K.L., Richman, M.B.: Euclidean distance as a similarity metric for principal component analysis. *Monthly Weather Review* **129** (2001) 540
21. Pampalk, E., Rauber, A., Merkl, D.: Using smoothed data histograms for cluster visualization in self-organizing maps. In: Proceedings of the Intl Conf on Artificial Neural Networks (ICANN 2002). (2002) 871–876