Uncovering the Hierarchical Structure of Text Archives by Using an Unsupervised Neural Network with Adaptive Architecture

Dieter Merkl and Andreas Rauber

Institut für Softwaretechnik, Technische Universität Wien
Favoritenstraße 9–11/188, A–1040 Wien, Austria
www.ifs.tuwien.ac.at/~dieter www.ifs.tuwien.ac.at/~andi

Abstract. Discovering the inherent structure in data has become one of the major challenges in data mining applications. It requires the development of stable and adaptive models that are capable of handling the typically very high-dimensional feature spaces. In this paper we present the *Growing Hierarchical Self-Organizing Map (GH-SOM)*, a neural network model based on the self-organizing map. The main feature of this extended model is its capability of growing both in terms of map size as well as in a three-dimensional tree-structure in order to represent the hierarchical structure present in a data collection. This capability, combined with the stability of the self-organizing map for high-dimensional feature space representation, makes it an ideal tool for data analysis and exploration. We demonstrate the potential of this method with an application from the information retrieval domain, which is prototypical of the high-dimensional feature spaces frequently encountered in today's applications.

1 Introduction

Today's information age may be characterized by constant massive production and dissemination of written information. More powerful tools for exploring, searching, and organizing the available mass of information are needed to cope with this situation. An attractive way to assist the user in document archive exploration is based on unsupervised artificial neural networks, especially self-organizing maps [3], for document space representation. A number of research publications show that this idea has found appreciation in the community [4, 5, 6, 7, 9, 13]. Self-organizing maps are used to visualize the similarity between documents in terms of distances within the two-dimensional map display. Hence, similar documents may be found in neighboring regions of the map.

Despite the large number of research reports on self-organizing map usage for document archive representation, some difficulties remain untouched. First, the determination of a suitable number of neurons requires some insight into the structure of the document archive. This cannot be assumed, however, in case of unknown document collections. Thus, it might be helpful if the neural network would be able to determine this number during its learning process. Second, hierarchical relations between the input data are not mirrored in a straight-forward manner. Obviously, we should expect such hierarchical relations in document collections where different subject matters are covered. The identification of these hierarchical relations remains a highly important data mining task that cannot be addressed conveniently within the framework of self-organizing map usage.

In order to overcome these two limitations of self-organizing maps we propose a novel neural network architecture in this paper, i.e. the *growing hierarchical self-organizing map*, *GH-SOM* for short. This neural network architecture is capable of determining the required number of units during its unsupervised learning process. Additionally, the data set is clustered hierarchically by relying on a layered architecture comprising a number of independent self-organizing maps within each layer.

The remainder of this paper is organized as follows. In Section 2 we provide an outline of architecture and learning rule of the *growing hierarchical self-organizing map*. Section 3 gives a description of the experimental data set, namely a collection of articles from the *Time Magazine*. We provide results from using both the self-organizing map and the *growing hierarchical self-organizing map* with this data set in Section 4. Related work is briefly described in Section 5. Finally, we present our conclusions in Section 6.

2 Growing hierarchical self-organizing maps

The key idea of the *growing hierarchical self-organizing map* (*GH-SOM*) is to use a hierarchical neural network structure composed of a number of individual layers each of which consists of independent *self-organizing maps* (*SOMs*). In particular, the neural network architecture starts with a single-unit *SOM* at layer 0. One *SOM* is used at layer 1 of the hierarchy. For every unit in this layer 1 map, a *SOM* might be added to the next layer of the hierarchy. This principle is repeated with the third and any further layers of the *GH-SOM*.

Since one of the shortcomings of *SOM* usage is its fixed network architecture in terms of the number of units and their arrangement, we rather rely on an incrementally growing version of the *SOM*. This relieves us from the burden of predefining the network's size which is now determined during the unsupervised training process according to the peculiarities of the input data space. Pragmatically speaking, the *GH-SOM* is intended to uncover the hierarchical relationship between input data in a straight-forward fashion. More precisely, the similarities of the input data are shown in increasingly finer levels of detail along the hierarchy defined by the neural network architecture. *SOMs* at higher layers give a coarse grained picture of the input data space whereas *SOMs* of deeper layers provide fine grained input discrimination. The growth process of the neural network is guided by the so-called *quantization error* which is a measure of the quality of input data representation.

The starting point for the growth process is the overall deviation of the input data as measured with the single-unit *SOM* at layer 0. This unit is assigned a weight vector m_0, $m_0 = [\mu_{0_1}, \mu_{0_2}, \ldots, \mu_{0_n}]^T$, computed as the average of all input data. The deviation of the input data, i.e. the *mean quantization error* of this single unit, is computed as given in Expression (1) with d representing the number of input data x. We will refer to the *mean quantization error* of a unit as **mqe** in lower case letters.

$$\mathbf{mqe}_0 = \frac{1}{d} \cdot ||m_0 - x|| \tag{1}$$

After the computation of $\mathbf{mqe}_0$, training of the *GH-SOM* starts with its first layer *SOM*. This first layer map initially consists of a rather small number of units, e.g. a grid of 2×2 units. Each of these units i is assigned an n-dimensional weight vector m_i, $m_i = [\mu_{i_1}, \mu_{i_2}, \ldots, \mu_{i_n}]^T$, $m_i \in \Re^n$, which is initialized with random values. It is important to note that the weight vectors have the same dimensionality as the input patterns.

The learning process of *SOMs* may be described as a competition among the units to represent the input patterns. The unit with the weight vector being closest to the presented input pattern in terms of the input space wins the competition. The weight vector of the winner as well as units in the vicinity of the winner are adapted in such a way as to resemble more closely the input pattern.

The degree of adaptation is guided by means of a learning-rate parameter α, decreasing in time. The number of units that are subject to adaptation also decreases in time such that at the beginning of the learning process a large number of units around the winner is adapted, whereas towards the end only the winner is adapted. These units are chosen by means of a neighborhood function h_{ci} which is based on the units' distances to the winner as measured in the two-dimensional grid formed by the neural network. In combining these principles of *SOM* training, we may write the learning rule as given in Expression (2), where x represents the current input pattern, and c refers to the winner at iteration t.

$$m_i(t+1) = m_i(t) + \alpha(t) \cdot h_{ci}(t) \cdot [x(t) - m_i(t)] \tag{2}$$

In order to adapt the size of this first layer *SOM*, the *mean quantization error* of the map is computed ever after a fixed number λ of training iterations as given in Expression (3). In this formula, u refers to the number of units i contained in the *SOM* m. In analogy to Expression (1), $\mathbf{mqe}_i$ is computed as the average distance between weight vector m_i and the input patterns mapped onto unit i. We will refer to the *mean quantization error* of a map as **MQE** in upper case letters.

$$\mathbf{MQE}_m = \frac{1}{u} \cdot \sum_i \mathbf{mqe}_i \tag{3}$$

The basic idea is that each layer of the *GH-SOM* is responsible for explaining some portion of the deviation of the input data as present in its preceding layer. This is done by adding units to the *SOMs* on each layer until a suitable size of the map is reached. More precisely, the *SOMs* on each layer are allowed to grow until the deviation present in the unit of its preceding layer is reduced to at least a fixed percentage τ_m. Obviously, the smaller the parameter τ_m is chosen the larger will be the size of the emerging *SOM*. Thus, as long as $\mathbf{MQE}_m \geq \tau_m \cdot \mathbf{mqe}_0$ holds true for the first layer map m, either a new row or a new column of units is added to this *SOM*. This insertion is performed neighboring the unit e with the highest *mean quantization error*, $\mathbf{mqe}_e$, after λ training iterations. We will refer to this unit as the *error unit*. The distinction whether a new row or a new column is inserted is guided by the location of the most dissimilar neighboring unit to the *error unit*. Similarity is measured in the input space. Hence, we insert a new row or a new column depending on the position of the neighbor with the most dissimilar weight vector. The initialization of the weight vectors of the new units is simply performed as the average of the weight vectors of the existing neighbors. After the insertion, the learning-rate parameter α and the neighborhood function h_{ci} are reset to their initial values and training continues according to the standard training process of *SOMs*. Note that we currently use the same value of the parameter τ_m for each map in each layer of the *GH-SOM*. It might be subject to further research, however, to search for alternative strategies, where layer or even map-dependent quantization error reduction parameters are utilized.

Consider Figure 1 for a graphical representation of the insertion of units. In this figure the architecture of the *SOM* prior to insertion is shown on the left-hand side where we find a map of 2×3 units with the *error unit* labeled by `e` and its most dissimilar neighbor signified by `d`. Since the most dissimilar neighbor belongs to another row within the grid, a new row is inserted between units `e` and `d`. The resulting architecture is shown on the right-hand side of the figure as a map of now 3×3 units.

Fig. 1. Insertion of units to a self-organizing map

As soon as the growth process of the first layer map is finished, i.e. $\mathbf{MQE}_m < \tau_m \cdot \mathbf{mqe}_0$, the units of this map are examined for expansion on the second layer. In particular, those units that have a large *mean quantization error* will add a new *SOM* to the second layer of the *GH-SOM*. The selection of these units is based on the *mean quantization error* of layer 0. A parameter τ_u is used to describe the desired level of granularity in input data discrimination in the final maps. More precisely, each unit i fulfilling the criterion given in Expression (4) will be subject to hierarchical expansion.

$$\mathbf{mqe}_i > \tau_u \cdot \mathbf{mqe}_0 \tag{4}$$

The training process and unit insertion procedure now continues with these newly established *SOMs*. The major difference to the training process of the second layer map is that now only that fraction of the input data is selected for training which is represented by the corresponding first layer unit. The strategy for row or column insertion as well as the termination criterion is essentially the same as used for the first layer map. The same procedure is applied for any subsequent layers of the *GH-SOM*.

The training process of the *GH-SOM* is terminated when no more units require further expansion. Note that this training process does not necessarily lead to a balanced hierarchy, i.e. a hierarchy with equal depth in each branch. Rather, the specific requirements of the input data is mirrored in that clusters might exist that are more structured than others and thus need deeper branching. Consider Figure 2 for a graphical representation of a trained *GH-SOM*. In particular, the neural network depicted in this figure consists of a single-unit *SOM* at layer 0, a *SOM* of 2×3 units in layer 1, six *SOMs* in layer 2, i.e. one for each unit in the layer 1 map. Note that each of these maps might have a different number and different arrangements of units as shown in the figure. Finally, we have one *SOM* in layer 3 which was expanded from one of the layer 2 units.

Fig. 2. Architecture of a trained *GH-SOM*

To summarize, the growth process of the *GH-SOM* is guided by two parameters τ_u and τ_m. The parameter τ_u specifies the desired quality of input data representation at the end of the training process. Each unit i with $\mathbf{mqe}_i > \tau_u \cdot \mathbf{mqe}_0$ will be expanded, i.e. a map is added to the next layer of the hierarchy, in order to explain the input data in more detail. Contrary to that, the parameter τ_m specifies the desired level of detail that is to be shown in a particular *SOM*. In other words, new units are added to a *SOM* until the **MQE** of the map is a certain fraction, τ_m, of the **mqe** of its preceding unit. Hence, the smaller τ_m the larger will be the emerging maps. Conversely, the larger τ_m the deeper will be the hierarchy.

3 Data Set

In the experiments presented hereafter we use the *TIME Magazine* article collection available at http://www.ifs.tuwien.ac.at/ifs/research/ir as a reference document archive. The collection comprises 420 documents from the *TIME Magazine* of the early 1960's. The documents can be thought of as forming topical clusters in the high-dimensional feature space spanned by the words that the documents are made up of. The goal is to map and identify those clusters on the 2-dimensional map display. Thus, we use full-text indexing to represent the various documents according to the vector space model of information retrieval. The indexing process identified 5923 content terms, i.e. terms used for document representation, by omitting words that appear in more than 90% or less than 1% of the documents. The terms are roughly stemmed and weighted according to a $tf \times idf$, i.e. term frequency $\times$ inverse document frequency, weighting scheme [14], which assigns high values to terms that are considered important in describing the contents of a document. Following the feature extraction process we end up with 420 vectors describing the documents in the 5923-dimensional document space, which are further used for neural network training.

4 Experimental Results

Figure 3 shows a conventional self-organizing map trained with the Times Article Collection data set. It consists of 10×15 units represented as table cells with a number of articles being mapped onto each individual unit. The articles mapped onto the same or neighboring units are considered to be similar to each other in terms of the topic they deal with. Due to space considerations we cannot present all the articles in the collection. We thus selected a number of units for detailed discussion.

We find, that the *SOM* has succeeded in creating a topology preserving representation of the topical clusters of articles. For example, in the lower left corner we find a group of units representing articles on the conflict in Vietnam. To name just a few, we find articles *T320, T369* on unit (14/1)[1], *T390, T418, T434* on unit (15/1) or *T390, T418, T434* on unit (15/2) dealing with the government crackdown on buddhist monks, next to a number of articles on units (15/4), (15/5) and neighboring ones, covering the fighting and suffering during the Vietnam War.

A cluster of documents covering affairs in the Middle-East is located in the lower right corner of the map around unit (15/10), next to a cluster on the so-called Profumo-Keeler affair, a political scandal in Great Britain in the 1960's, on and around units (11/10) and (12/10). Above this area, on units (6/10) and neighboring ones we find articles on elections in Italy and possible coalitions, next to two units (3/10) and (4/10) covering elections in India. Similarly, all other units on the map can be identified to represent a topical cluster of news

[1] We use the notion (x/y) to refer to the unit located in row x and column y of the map, starting with (1/1) in the upper left corner

Fig. 3. 10×15 *SOM* of the *Time Magazine* collection

articles. For a more detailed discussion of the articles and topic clusters found on this map, we refer to [12] and the online-version of this map available at http://www.ifs.tuwien.ac.at/ifs/research/ir.

While we find the *SOM* to provide a good topologically ordered representation of the various topics found in the article collection, no information about topical hierarchies can be identified from the resulting flat map. Apart from this we find the size of the map to be quite large with respect to the number of topics identified. This is mainly due to the fact that the size of the map has to be determined in advance, before any information about the number of topical clusters is available.

To overcome these shortcomings we trained a growing hierarchical SOM. Based on the artificial unit representing the means of all data points at layer 0, the GH-SOM training algorithm started with a 2×2 SOM at layer 1. The training process for this map continued with additional units being added until the quantization error fell below a certain percentage of the overall quantization error of the unit at layer 0. The resulting first-layer map is depicted in Figure 4. The map has grown for two stages, adding one row and one column respectively, resulting in 3×3 units representing 9 major topics in the document collection.

For convenience we list the topics of the various units, rather then the individual articles in the figure. For example, we find unit (1/1) to represent all articles related to the situation in Vietnam, whereas Middle-East topics are cov-

Fig. 4. Layer 1 of the *GH-SOM*

ered on unit (1/3), or articles related to elections and other political topics on unit (3/1) in the lower left corner to name but a few.

Based on this first separation of the most dominant topical clusters in the article collection, further maps were automatically trained to represent the various topics in more detail. This results in 9 individual maps on layer 2, each representing the data of the respective higher-layer unit in more detail. Some of the units on these layer 2 maps were further expanded as distinct *SOMs* in layer 3.

The resulting layer 2 maps are depicted in Figure 5. Please note, that – according to the structure of the data – the maps on the second layer have grown to different sizes, such as a small 2×2 map representing the articles of unit (3/1) of the first map, up to 3×3 maps for the units (2/1), (3/2) and (3/3). Taking a more detailed look at the first map of layer 2 representing unit (1/1) of layer 1 we find it to give a clearer representation of articles covering the situation in Vietnam. Units (1/1) and (2/1) on this map represent articles on the fighting during the Vietnam War, whereas the remaining units represent articles on the internal conflict between the catholic government and buddhist monks. At this layer, the two units (1/2) and (3/2) have further been expanded to form separate maps with 3×3 units each at layer 3. These again represent articles on the war and the internal situation in Vietnam in more detail.

To give another example of the hierarchical structures identified during the *growing hierarchical SOM* training process, we may take a look at the 2×3

Fig. 5. Layer 2 of the *GH-SOM*: 1 *SOM* per unit of layer 1 *SOM*

map representing the articles of unit (3/1) of the first layer map. All of these articles were found to deal with political matters on layer 1. This common topic is now displayed in more detail at the resulting second-layer map. For example, we find unit (1/3) to represent articles on the elections in India. Next to these, we find on units (1/2) and (2/3) articles covering the elections and discussions about political coalitions between socialists and christian democrats in Italy. The remaining 3 units on this map deal with different issues related to the Profumo-Keeler scandal in Great Britain, covering the political hearings in parliament as well as background information on this scandal and the persons involved. Again, some of the units have been expanded at a further level of detail forming 3×2 or 3×3 SOMs on layer 3.

For comparing the *GH-SOM* with its flat counterpart we may identify the locations of the articles on the 9 second-layer maps on the corresponding 10×15 *SOM*. This allows us to view the hierarchical structure of the data on the flat map. We find that, for example, the cluster on Vietnam simply forms one larger coherent cluster on the flat map in the lower left corner of the map covering the rectangle spanned by the units (14/1) and (15/5). The same applies to the cluster of Middle-East affairs, which is represented by the map of unit (1/3) in the *growing hierarchical SOM*. This cluster is mainly located in the lower right corner of the flat *SOM*. The cluster of political affairs, represented by unit (3/1) on the first layer of the *GH-SOM* and represented in more detail on its subsequent layers, is spread across the right side of the flat *SOM*, covering more or less all units on columns 9 and 10 and between rows 3 and 12. Note, that this common topic of political issues is not easily discernible from the overall map representation in the flat *SOM*, where exactly this hierarchical information is lost. The subdivision of this cluster on political matters becomes further evident when we consider the second layer classification of this topic area, where the various sub-topics are clearly separated, covering Indian elections, Italian coalitions and the British Profumo-Keeler scandal.

As another interesting feature of the *GH-SOM* we want to emphasize on is the overall reduction in map size. During analysis we found the second layer of the *GH-SOM* to represent the data at about the same level of topical detail as the corresponding flat *SOM*. Yet the number of units of all individual second-layer *SOMs* combined is only 87 as opposed to 150 units in the flat 10×15 *SOM*. Of course we might decide to train a smaller flat *SOM* of, say 9×10 units. However, with the *GH-SOM* model, this number of units is determined automatically, and only the necessary number of units is created for each level of detail representation required by the respective layer. Furthermore, not all branches are grown to the same depth of the hierarchy. As can be seen from Figure 5, only some of the units are further expanded in a layer 3 map. With the resulting maps at all layers of the hierarchy being rather small, activation calculation and winner evaluation is by orders of magnitude faster than in the conventional model. Apart from the speedup gained by the reduced network size, orientation for the user is highly improved as compared to the rather huge maps which can not be easily comprehended as a whole.

5 Related work

A number of extensions and modifications have been proposed over the years in order to enhance the applicability of *SOMs* to data mining, specifically cluster identification. Some of the approaches, such as the *U-Matrix* [15], or the *Adaptive Coordinates* and *Cluster Connection* techniques [8] focus on the detection and visualization of clusters in conventional *SOMs*. Similar cluster information can also be obtained using our *LabelSOM* method [11], which automatically describes the characteristics of the various units. Grouping units that have the same descriptive keywords assigned to them allows to identify topical clusters within the *SOM* map area. However, none of the methods identified above facilitates the detection of hierarchical structure inherent in the data.

The *hierarchical feature map* [10] addresses this problems by modifying the *SOM* network architecture. Instead of training a flat *SOM* map, a balanced hierarchical structure of *SOMs* is trained. Similar to our *GH-SOM* model, the data mapped onto one single unit is represented at some further level of detail in the lower-level map assigned to this unit. However, this model rather pretends to represent the data in a hierarchical way rather than really reflecting the structure of the data. This is due to the fact that the architecture of the network has to be defined in advance, i.e. the number of layers and the size of the maps at each layer is fixed prior to network training. This leads to the definition of a balanced tree which is used to represent the data. What we want, however, is a network architecture definition based on the actual data presented to the network. This requires the *SOM* to actually use the data available to define its architecture, the required levels in the hierarchy and the size of the map at each level, none of which is present in the *hierarchical feature map* model.

The necessity of having to define the size of the *SOM* in advance has been addressed in several models, such as the *Incremental Grid Growing* [1] or *Growing Grid* [2] models. The latter, similar to our *GH-SOM* model, adds rows and columns during the training process, starting with an initial 2×2 SOM. However, the main focus of this model lies with an equal distribution of input signals across the map, adding units in the neighborhood of units that represent an unproportionally high number of data points. It does thus not primarily reflect the concept of representation at a certain level of detail, which is rather expressed in the overall quantization error rather then the number of data points mapped onto certain areas. The *Incremental Grid Growing* model, on the other hand, can add new units only on the borders of the map. Neither of this models, however, takes the inherently hierarchical structure of data into account.

6 Conclusions

We have presented the *Growing Hierarchical Self-Organizing Map* (*GH-SOM*), a neural network based on the self-organizing map (*SOM*), a model that has proven to be effective for cluster analysis of very high-dimensional feature spaces. Its main benefits are due to the model's capabilities to (1) determine the number of neural processing units required in order to represent the data at a desired level of detail and (b) to create a network architecture reflecting the hierarchical structure of the data. The resulting benefits are numerous: first, the processing time is largely reduced by training only the necessary number of units for a certain degree of detail representation. Second, the *GH-SOM* by its very architecture resembles the hierarchical structure of data, allowing the user to understand and analyze large amounts of data in an explorative way. Third, with the various emergent maps at each level in the hierarchy being rather small, it is easier for the user to keep an overview of the various clusters identified in the data and to build a cognitive model of it in a very high-dimensional feature space. We have demonstrated the capabilities of this approach by an application from the information retrieval domain, where text documents, which are located in a high-dimensional feature space spanned by the words in the documents, are clustered by their mutual similarity and where the hierarchical structure of these documents is reflected in the resulting network architecture.

References

1. J. Blackmore and R. Miikkulainen. Incremental grid growing: Encoding high-dimensional structure into a two-dimensional feature map. In *Proc. of the IEEE Int'l. Conf. on Neural Networks (ICNN'93)*, San Francisco, CA, USA, 1993.
2. B. Fritzke. Growing grid – a self-organizing network with constant neighborhood range and adaption strength. *Neural Processing Letters*, 2, No. 5:1 – 5, 1995.
3. T. Kohonen. *Self-Organizing Maps.* Springer Verlag, Berlin, Germany, 1995.
4. T. Kohonen. Self-organization of very large document collections: State of the art. In *Proc Int'l Conf on Artificial Neural Networks*, Skövde, Sweden, 1998.
5. X. Lin, D. Soergel, and G. Marchionini. A self-organizing semantic map for information retrieval. In *Proc. Int'l ACM SIGIR Conf. on R & D in Information Retrieval*, Chicago, IL, 1991.
6. D. Merkl. Text classification with self-organizing maps: Some lessons learned. *Neurocomputing*, 21(1–3), 1998.
7. D. Merkl. Text data mining. In *A Handbook of Natural Language Processing: Techniques and Applications for the Processing of Language as Text.* Marcel Dekker, New York, 1998.
8. D. Merkl and A. Rauber. Alternative ways for cluster visualization in self-organizing maps. In *Proc. of the Workshop on Self-Organizing Maps (WSOM97)*, Helsinki, Finland, 1997.
9. D. Merkl and A. Rauber. Uncovering associations between documents. In *Proc. International Joint Conference on Artificial Intelligence (IJCAI99)*, Stockholm, Sweden, 1999.
10. R. Miikkulainen. Script recognition with hierarchical feature maps. *Connection Science*, 2:83 – 101, 1990.
11. A. Rauber and D. Merkl. Automatic labeling of self-organizing maps: Making a treasure map reveal its secrets. In *Proc. 4th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD99)*, Beijing, China, 1999. Springer Verlag.
12. A. Rauber and D. Merkl. Using self-organizing maps to organize document collections and to characterize subject matters: How to make a map tell the news of the world. In *Proc. 10th Int'l Conf. on Database and Expert Systems Applications (DEXA99)*, Florence, Italy, 1999.
13. D. Roussinov and M. Ramsey. Information forage through adaptive visualization. In *Proc. ACM Conf. on Digital Libraries 98 (DL98)*, Pittsburgh, PA, USA, 1998.
14. G. Salton. *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer.* Addison-Wesley, Reading, MA, 1989.
15. A. Ultsch. Self-organizing neural networks for visualization and classification. In *Information and Classification. Concepts, Methods and Application.* Springer Verlag, 1993.

.

Dieter Merkl, Andreas Rauber: **Uncovering the Hierarchical Structure of Text Archives by Using an Unsupervised Neural Network with Adaptive Architecture**.
In: Proceedings of the 4. Pacific-Asia Conference on Knowledge Discovery and Data Mining, April 18. - 20. 2000, Kyoto, Japan, Springer Lecture Notes in Artificial Intelligence (LNCS-LNAI 1805), pp. 384 - 395, Springer, Germany.

This article was processed using the LaTeX macro package with LLNCS style