Chapter 14

Other projects

14.1 Adaptive committee techniques

Matti Aksela, Jorma Laaksonen, Erkki Oja

Combining the results of several classifiers can improve performance because in the outputs of the individual classifiers the errors are not necessarily overlapping. Also the combination method can be adaptive. The two most important features of the member classifiers that affect the committee's performance are their individual error rates and the diversity of the errors. The more different the mistakes made by the classifiers, the more beneficial the combination of the classifiers can be.

Selecting member classifiers is not necessarily simple. Several methods for classifier diversity have been presented to solve this problem. In [1] a scheme weighting similar errors made in an exponential fashion, the Exponential Error Count method, was found to provide good results. Still, the best selection of member classifiers is highly dependent on the combination method used.

We have experimented with several adaptive committee structures. Two effective methods have been the Dynamically Expanding Context (DEC) and Class-Confidence Critic Combining (CCCC) schemes [2]. The DEC algorithm was originally developed for speech recognition purposes. The main idea is to determine just a sufficient amount of context for each individual segment so that all conflicts in classification results can be resolved.

In our CCCC approach the main idea is to try to produce as good as possible an estimate on the classifier's correctness based on its prior behavior for the same character class. This is accomplished by the use of critics that assign a confidence value to each classification. The confidence value is obtained through constructing and updating distribution models of distance values from the classifier for each class in every critic. These distribution models are then used to extract the needed confidence value, based on prior results in addition to the sample being processed. The committee then uses a decision mechanism to produce the final output from the input label information and critic confidence values. The adaptive committee structures have been shown to be able to improve significantly on their members' results [2].

Also classifiers that are adaptive in themselves can be combined using an adaptive committee, and recent experiments have shown that this adaptive combination of adaptive classifiers can produce even better results than either method alone. Combining adaptive classifiers is feasible through the use of a weighting scheme to obtain better balance between the use of prior information and robustness under changing conditions [3].

References

[1] Matti Aksela and Jorma Laaksonen. Using diversity of errors for selecting members of a committee classifier. *Pattern Recognition*, 39(4):608–623, 2006.

[2] Matti Aksela, Ramūnas Girdziušas, Jorma Laaksonen, Erkki Oja, and Jari Kangas. Methods for adaptive combination of classifiers with application to recognition of handwritten characters. *International Journal of Document Analysis and Recognition*, 6(1):23–41, 2003.

[3] Matti Aksela and Jorma Laaksonen. On adaptive confidences for critic-driven classifier combining. In *Proceedings of International Conference on Advances in Pattern Recognition*, volume 2, pages 71–80, 2005.

14.2 Data analysis using the Evolving Tree

Jussi Pakkanen

Modern data analysis problems usually have to deal with very large databases. When the amount of data samples grow to millions or tens of millions, many traditional tools and techniques slow down noticeably. This, combined with the curse of dimensionality, makes problems involving large data sets very difficult to approach.

Our research has focused on finding novel methods to combine neural network systems with large data set manipulation tools of computer science. The goal is to create new neural systems that can be used to analyze huge data bases efficiently while retaining a high precision. We propose the *The Evolving Tree* [1] for this task.

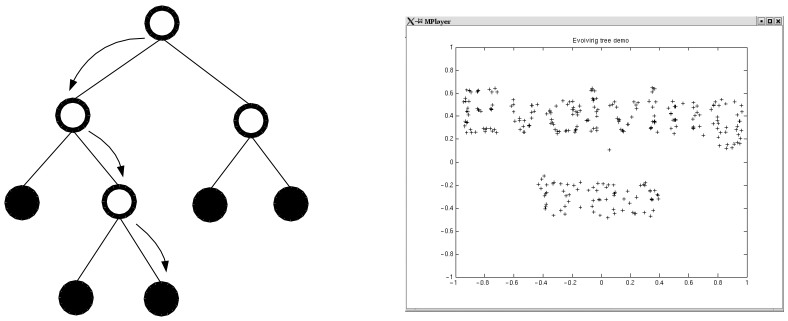

Figure 14.1: The general architecture of the Evolving Tree and an example of adaptation to data.

Figure 14.1 demonstrates the basic properties of the Evolving Tree. The left image shows how the tree is made of two kinds of nodes. The black *leaf nodes* are the actual data analysis nodes, which perform vector coding. The white *trunk nodes* form an efficient search tree to the leaf nodes. The arrows show how a single search on the tree might progress. During training the Evolving Tree grows by creating new leaf nodes to those areas of the data space that are deemed to be underrepresented.

The right image on Figure 14.1 shows how the Evolving Tree adapts to an artificial two-dimensional data set. The dots are the code vectors.. The training had started with a single node, but the tree has grown in size to better explain the data.

Tests on real world industrial data shows that ETree performs favorably when compared to other similar methods [2].

References

[1] Jussi Pakkanen, Jukka Iivarinen and Erkki Oja The Evolving Tree — A Novel Self-Organizing Network for Data Analysis, in *Neural Processing Letters*, Volume 20, Issue 3, pages 199–211, December 2004.

[2] Jussi Pakkanen and Jukka Iivarinen, Analyzing Large Image Databases with the Evolving Tree, in *Proceedings of the Third International Conference on Advances in Pattern Recognition, ICAPR05*, LNCS 3686, pages 192–198, August 2005.

14.3 Independent variable group analysis

Krista Lagus, Antti Honkela, Jeremias Seppä, Paul Wagner

Independent variable group analysis (IVGA) [1] is a principle for grouping observed input variables so that mutual dependences between variables are strong within a group and weak between groups.

In problems with a large number of diverse observations there are often groups of input variables that have strong mutual dependences within the group but which can be considered practically independent of the input variables in other groups. It can be expected that the larger the problem domain, the more independent groups there are. Estimating a model for each independent group separately produces a more compact representation than applying the model to the whole set of variables. Compact representations are computationally beneficial and, moreover, offer better generalization.

Usually such variable grouping is performed by a domain expert, prior to modeling with automatic, adaptive methods. As expert knowledge may be unavailable, or expensive and time-consuming, automating the task can considerably save resources. The IVGA is a practical, efficient and general approach for obtaining compact representations that can be regarded as sparse codes, as well.

The IVGA project is a collaboration with Dr. Esa Alhoniemi (University of Turku) and Dr. Harri Valpola (Helsinki University of Technology, Laboratory of Computational Engineering).

The IVGA algorithm

Any IVGA algorithm consists of two parts, (1) grouping of variables, and (2) construction of a separate model for each variable group. An independent variable grouping is obtained by comparing models with different groupings using a suitable cost function. In principle any model can be used, if the necessary cost function is derived for the model family.

A practical grouping algorithm for implementing the IVGA principle was first presented in [1]. The method used vector quantizers (VQs) learned with variational Bayesian methods [2] to model the individual groups.

Recent development of IVGA [3] has concentrated on extending our algorithmic implementation to handle mixed data consisting of both real valued and nominal variables. A public software package is in preparation and new experiments have been made. In order to allow both real and nominal variables, the vector quantizers were replaced with mixture models so that the mixture components were Gaussians in the real valued case and multinomial distributions in the nominal case.

Experimental results

In this experiment, IVGA was used to group the variables representing attributes of components required by an assembly robot in mounting of the components on a printed circuit board. Finding correct settings for the attributes by hand is difficult and association rules have been applied to model their dependences [4]. Extraction of the rules from the data is computationally heavy, and memory consumption of the data structure (for example, a trie) for the rules is very high for large data sets. Splitting the variables to weakly dependent groups decreases the complexity significantly.

The training data used in this experiment consisted of 24 attributes (17 nominal, 7 real valued) with values for 1000 components. Computation tiems, memory consumption,

and prediction accuracy for a separate testing data set using association rules built for the full data set and for the three groups discovered by IVGA are presented in Table 14.1. Splitting of the data using IVGA lead to significant improvements in the efficiency of the obtained model: it accelerated computation of the rules, dramatically reduced the size of the data structure, and decreased the number of the incorrect predictions. On the other hand, the number of missing predictions was clearly larger for the grouped data than for the whole data, because for the first attribute value of every group, no prediction could be made whereas for the whole data, only the prediction for the first attribute could not be obtained [3].

	Whole data	**Grouped data**
Computation time (s)	194	< 1
Size of trie (nodes)	1 054 512	3 914
Correct predictions (%)	38.05	32.45
Incorrect predictions (%)	1.61	0.68
Missing predictions (%)	60.43	66.88

Table 14.1: Summary of the results of the component data experiment. All the quantities for the grouped data are sums over the three groups. Also note that the size of trie is the same as the number of association rules.

In conclusion, the experimental results show that it is worthwhile to group variables according to independence, and that the presented algorithm is able to do this and in doing so, obtains more compact models.

References

[1] K. Lagus, E. Alhoniemi, and H. Valpola, "Independent variable group analysis," in *International Conference on Artificial Neural Networks - ICANN 2001*, ser. LLNCS, G. Dorffner, H. Bischof, and K. Hornik, Eds., vol. 2130. Vienna, Austria: Springer, August 2001, pp. 203–210.

[2] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul, "An introduction to variational methods for graphical models," in *Learning in Graphical Models*, M. Jordan, Ed. Cambridge, MA, USA: The MIT Press, 1999, pp. 105–161.

[3] K. Lagus, E. Alhoniemi, J. Seppä, A. Honkela, and P. Wagner, "Independent variable group analysis in learning compact representations for data," in *Proceedings of the International and Interdisciplinary Conference on Adaptive Knowledge Representation and Reasoning (AKRR'05)*, T. Honkela, V. Könönen, M. Pöllä, and O. Simula, Eds., Espoo, Finland, June 2005, pp. 49–56.

[4] E. Alhoniemi, T. Knuutila, M. Johnsson, J. Röyhkiö, and O. S. Nevalainen, "Data mining in maintenance of electronic component libraries," in *Proceedings of the IEEE 4th International Conference on Intelligent Systems Design and Applications*, 2004, vol. 1, pp. 403–408.

14.4 Worldwide research on and using the Self-Organizing Map

Teuvo Kohonen, Timo Honkela and Matti Pöllä

The Self-Organizing Map [2] (SOM) is an effective method for the analysis and visualization of high-dimensional data. It converts complex, nonlinear statistical relationships between high-dimensional data items into simple geometric relationships on a low-dimensional display. As it thereby compresses information while preserving the most important topological and metric relationships of the primary data items on the display, it may also be thought to produce some kind of abstractions. These two aspects, visualization and abstraction, can be utilized in a number of ways in complex tasks.

The Self-Organizing Map has attracted a great deal of interest among researches and practitioners in a wide variety of fields. The SOM has been analyzed extensively, a number of variants have been developed and, perhaps most notably, it has been applied extensively within fields ranging from engineering sciences to medicine, biology, and economics. Comprehensive lists of scientific papers that use the algorithms, have benefited from them, or contain analyses of them has earlier been collected covering the time until early 2002 [3, 4]. A new, for the moment unpublished update of the bibliography is under preparation (by M. Pöllä, T. Honkela and T. Kohonen). Based on this newest collection, we can report that by the end of 2005, there were altogether approximately 7000 references.

The first two scientific events dedicated to the SOM, its theory and applications, the Workshop on Self-Organizing Maps, were organized at the Helsinki University of Technology in 1997 and 1999. Since then, the WSOM has been organized biannually (in 2001 in Lincoln, England, in 2003 in Hibikino, Kitakyushu, Japan and in 2005 in Paris by University Paris 1). In addition, the SOM is often a specific theme in conferences related both to computer science and to the several application areas of the SOM ranging from industry to bioinformatics.

In summary, the research on the Self-Organizing map is very active still more than 20 years after its original invention and publication [1]. New results on its theory, extensions and applications are published in practice on daily basis. The practical importance of the SOM can also be highlighted by the fact that the Self-Organizing Map has been included in a large number of commercial analytical software packages both by large companies producing general purpose tools or by companies dedicated to the development of SOM-based tools and applications .

In the following, we describe a novel development, a model called Self-Organizing Neural Projection and give an overview on the body of research in our laboratory based on the Self-Organizing Map.

References

[1] T. Kohonen (1982). Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43:59-69.

[2] T. Kohonen (2001). *Self-Organizing Maps*. Third, extended edition. Springer.

[3] S. Kaski, J. Kangas, T. Kohonen (1998). Bibliography of self-organizing map (SOM) papers: 1981-1997. *Neural Computing Surveys*, 1: 102-350.

[4] M. Oja, S. Kaski, and T. Kohonen (2003). Bibliography of Self-Organizing Map (SOM) Papers: 1998-2001 Addendum. *Neural Computing Surveys*, 3: 1-156.

14.5 Self-Organizing Neural Projections

Teuvo Kohonen

The SOM algorithm was developed for the creation of abstract-feature maps [3]. It has been accepted widely as a data-mining tool, and the principle underlying it may also explain how the feature maps of the brain are formed [4]. However, it is not correct to use this algorithm for a model of pointwise neural projections such as the somatotopic maps or the maps of the visual field, first of all, because the SOM does not transfer signal patterns: the winner-take-all function at its output only defines a singular response. Neither can the original SOM produce superimposed responses to superimposed stimulus patterns.

The SOM was not the first model of cortical organization (cf., e.g., the line detector model of v.d. Malsburg [6], and Amari's "synaptic field" model of the Type 1 pointwise maps [1]. Unfortunately, none of these attempts was a success. For instance, Amari's maps were not globally ordered. They were always parcelled into small, ordered patches, between which the ordering changed abruptly. On the other hand, v.d. Malsburg's model was "brittle," because the reported ordering only took place for a parameter value that was defined by three decimal places, and the maps could not be generalized.

The reason for the failure of the earlier models was that they were solely based on excitatory and inhibitory lateral connections, and the Hebbian rule of synaptic plasticity. By means of the lateral connections, the output activity was first clustered spatially, and the adaptation then took place in these clusters in proportion to input and output activities. However, when using these models, the activity clusters, in order to obtain globally ordered maps, should have been very wide, of the same order of magnitude as the dimension of the array. But then one could have hardly regarded such wide activity clusters as pointwise output responses. On the other hand, if the clusters had been made smaller, the maps would only have been organized into small, disjoint local patches. Also, such clustering would have disturbed the transfer of the original signals.

In the biological realms, genetic information defines a very rough initial order of the neural projections. Refinement of this order begins already prenatally, by means of endogenous signals generated by the network itself. The final resolution of the mapping, and optimization of the neural resources (magnification factor), however, are achieved postnatally, according to the sensory experiences. It has been demonstrated that exposing newborn rats to continuous moderate-level acoustic noise, the development and refinement of the tonotopic maps will be delayed long beyond normal periods [2]. It has also been shown that the exposure of infant rats to complex tone sequences results in altered auditory cortex organization [7]. These observations prove that the input-driven organization of the brain maps is a fact and needs a new theoretical model.

We have recently introduced a novel self-organizing system model related to the SOM that has a linear transfer function for patterns and combinations of patterns all the time [5]. Starting from a randomly interconnected pair of neural layers, and using random mixtures of patterns for training, it creates a pointwise-ordered projection from the input layer to the output layer. If the input layer consists of feature detectors, the output layer forms a feature map of the inputs. More detailed description of the model can be found in [5].

References

[1] Amari, S.-I. (1980). Topographic organization of nerve fields. *Bulletin of Mathematical Biology*, 42:339–364.

[2] Chang, E.F. and Merzenich, M.M. (2003). Environmental Noise Retards Auditory Cortical Development *Science*, 300 (5618): 498–502.

[3] T. Kohonen (1982). Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43:59-69.

[4] Kohonen, T. (2001). *Self-Organizing Maps*. Third, extended edition. Springer.

[5] Kohonen, T. (2005). Pointwise organizing projections. M. Cottrell (ed.), *Proceedings of the 5th Workshop on Self-Organizing Maps*, Paris, pp. 1-8.

[6] von der Malsburg, Ch. (1973). Self-organization of orientation sensitive cells in the striate cortex. *Kybernetik*, 14, 85–100.

[7] Nakahara, H. Zhang, L.I. and Merzenich, H.H. (2004). Specialization of primary auditory cortex processing by sound exposure in the "critical period". PNAS 101: 7170–7174

14.6 Applications of the Self-Organizing Map

Teuvo Kohonen, Olli Simula, Erkki Oja, Samuel Kaski, Timo Honkela, Jukka Iivarinen, Markus Koskela, Mikko Kurimo, Jorma Laaksonen, Krista Lagus, Kimmo Raivio, Miki Sirola

The Self-Organizing Map is applied in several areas of research in our laboratory. The application areas incluce:

- content-based image retrieval (see Chapter 7.1 for a detailed description),
- modeling the emergence of cognitive and conceptual representations (Chapter 8.2),
- language modeling (Chapters 9.3 and 10.3) and word sense disambiguation using document maps (Chapter 10.2), and
- multiple applications in the area of intelligent data engineering including the analysis of mobile radio access network (Chapter 11.2), decision support (Chapter 11.4), and modeling dependencies in data (Chapter 11.5).

In addition, a new hierarchical self-organizing system called the Evolving Tree has been described in Chapter 14.2.