Self-Organizing Systems for Knowledge Discovery in Large Databases

William H. Hsu, Loretta S. Auvil, William M. Pottenger, David Tcheng, and Michael Welge
{ bhsu | lauvil | billp | dtcheng | welge }@ncsa.uiuc.edu
http://www.ncsa.uiuc.edu/People/{ bhsu | lauvil | billp | dtcheng | welge }
Automated Learning Group (http://www.ncsa.uiuc.edu/STI/ALG)
National Center for Supercomputing Applications, University of Illinois at Urbana-Champaign
152 Computing Applications Building, 605 East Springfield Avenue, Champaign, IL 61820

Abstract

We present a framework in which self-organizing systems can be used to perform change of representation *on knowledge discovery problems, to learn from very large databases. Clustering using self-organizing maps is applied to produce multiple,* intermediate *training targets that are used to define a new supervised learning and mixture estimation problem. The input data is partitioned using a state space search over subdivisions of attributes, to which self-organizing maps are applied to the input data* as restricted to a subset *of input attributes. This approach yields the variance-reducing benefits of techniques such as stacked generalization, but uses self-organizing systems to discover* factorial *(modular) structure among abstract learning targets. This research demonstrates the feasibility of applying such structure in very large databases to build a mixture of ANNs for data mining and KDD. Areas of applications include multi-attribute risk assessment using insurance policy data, text document categorization, and anomaly detection.*

Keywords: large-scale data mining, knowledge discovery, very large databases, clustering, self-organizing maps

Introduction

This paper presents several applications of self-organizing systems to problems of knowledge discovery in very large databases. The purpose of self-organization in these problems is to perform *change of representation* for supervised learning, thereby reducing the computational complexity of the learning problem given the transformed problem. Cluster formation using self-organizing maps is applied to produce multiple, *intermediate* training targets [1] (synthetic attributes) that are used to define a new supervised learning and mixture estimation problem.

The input data is first partitioned using a state space search over subdivisions of attributes (this approach is an extension of existing work on attribute subset selection). Self-organizing maps are applied to the input data in order to formulate learning targets based on the *data as restricted to each subset* of input attributes. This approach yields the variance-reducing benefits of techniques such as stacked generalization [2], but facilitates the use of *factorial structure* (the ability of abstract targets to be factored). This research demonstrates the feasibility of learning factorial structure from data (using prior knowledge about the problem to partition the input), and of applying this structure in policy data to build a mixture model composed of ANNs.

Areas of applications include multi-attribute risk assessment (prediction of expected financial loss) using insurance policy data, text document categorization, and anomaly (fraud, intrusion, and crisis) detection. This paper presents case studies in each area, based on current research projects.

Background

Decomposition of Learning Tasks by Clustering

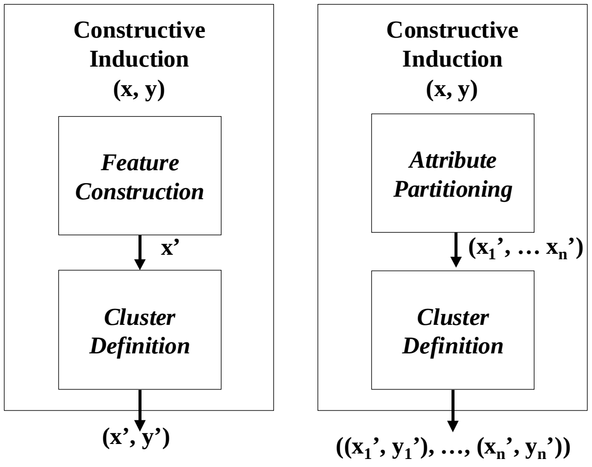


Figure 1. Role of cluster definition in two alternative constructive induction schemes

Figure 1 depicts two methods for unsupervised learning (constructive induction), both of which apply clustering algorithms to one or more subsets of input attributes (data channels) to achieve *change of representation* [3] for a supervised learning task.

The following is a brief introduction to learning task decomposition by attribute partitioning [4]. *Attribute subset selection* is the task of focusing a learning algorithm's attention on some subset of the given input attributes, while ignoring the rest [5, 6]. In this research, subset selection is adapted to the systematic decomposition of learning problems over heterogeneous time series. Instead of focusing a single algorithm on a single subset, the set of all input attributes is partitioned, and a specialized algorithm is focused on *each* subset. While subset selection is designed for refinement of attribute sets for single-model learning, attribute partitioning is designed specifically for multiple-model learning. This new approach adopts the role of feature construction in constructive induction: to formulate a new input specification from the original one [7], as depicted on the right hand side of Figure 1. It uses subset partitioning to *decompose* a learning task into parts that are individually useful, using *aggregation* of attributes (the $\mathbf{x}_i$). By contrast, attribute subset selection attempts to *reduce* attributes to a single useful group. This permits multiple-model methods such as *bagging* [8], *boosting* [9], and *hierarchical mixture models* [10] to be adapted to multistrategy learning [4].

Partitioning permits new intermediate concepts (the $\mathbf{y}_i$) to be formed by unsupervised learning (e.g., conceptual clustering [11] or cluster formation using self-organizing algorithms [12, 13]). The newly defined problem or problems can then be mapped to one or more appropriate hypothesis languages (model specifications). In our new system, the subproblem definitions obtained by partitioning of attributes also specify a mixture estimation problem (i.e., data fusion step occurs after training of the models for all the subproblems). [4] describes a metric-based model selection algorithm for this architecture.

One significant benefit of this abstraction approach is that it exploits factorial structure in abstract (decomposable) learning tasks. This results in a reduction in network complexity compared to non-modular or non-hierarchical methods, *whenever this structure can be identified* using prior knowledge or through clustering and vector quantization methods, as discussed in this paper. In addition, the bottom-up construction supports natural grouping of input attributes based on *modalities* of perception (e.g., the data *channels* or observable attributes available to each "specialist" via a particular sensor) [13]. Finally, experiments demonstrate that the achievable test error on decomposable time series learned using a specialist-moderator network is lower than that for non-modular feedforward or temporal ANN, when both are trained to convergence.

Role of Neural Clustering in KDD

In addition to Kohonen's Self-Organizing Feature Map (SOFM or SOM, a self-organizing algorithm first presented in [13] is used to organize each of the training examples into self-organized equivalence classes (SOECs). The latter algorithm was chosen to produce only a crude measure of statistical proximity and intentionally does not apply any "high-powered" clustering technique.

The cluster definition step in each input vector having an assigned target class for training the relevant expert, or specialist network (each one being an ANN component: a multilayer perceptron, simple recurrent network, time-delay neural network, or Gamma memory [14, 13]). *k*-means clustering, Gaussian clustering algorithms, and structured competitive clustering algorithms such as hierarchical agglomeration and "neural trees" [15] may also be used in this cluster definition step, depending on the *partition evaluation metric* [16].

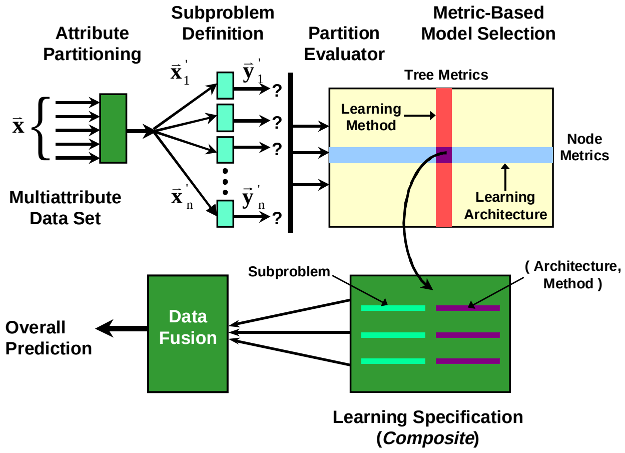


Figure 2. A framework for composite learning

Figure 2 depicts a complete learning system [16] for multi-attribute data sets that exhibit factorial (modular) structure, such as heterogenous time series, which arise from multiple data sources. The central elements of this system are: *attribute partitioning*, *metric-based model selection*, and a *data fusion* mechanism for integration of multiple models. Given a specification for reformulated (reduced or partitioned) input, the new intermediate concepts $\vec{\mathbf{y}}_i'$ can be formed by cluster definition; the newly defined problem or problems can then be mapped to one or more appropriate hypothesis languages (model specifications). [16] presents **Select-Net**, a high-level algorithm for generating this specification, which we shall refer to as a *composite*; we refer the reader to [4] two specific experiments demonstrating composite learning.

The **Select-Net** algorithm also configures and trains subnetworks in a hierarchical mixture (whose components may include inducers other than ANNs [17, 18]); a data fusion step occurs after individual training of each model. The system incorporates attribute partitioning into constructive induction to obtain multiple problem definitions (decomposition of learning tasks); applies metric-based model selection over subtasks to *search for efficient hypothesis preferences*; and integrates these techniques in a data fusion (mixture estimation) framework.

Methodology

Classification Problems in Large-Scale KDD

This section presents a short synopsis of the applications first in order to elucidate the data preparation steps.

Four applications of this research currently employ SOM and related topology-preserving projection algorithms. The first is classification of insurance policy records, a discretized prediction problem over coarse-grained time series. The second is technical text document categorization, a clustering problem that requires a very-high-dimension projection (for cluster formation) as well as sophisticated segmentation and labeling algorithms. The third is multisensor integration in order to predict hazardous and potentially catastrophic conditions from historical (time series) training data and a continuation of its observable (input) component. This prediction task is also known as *crisis monitoring*, a form of pattern recognition that is useful in decision support (or *recommender* [6]) systems for many time-critical applications. These include crisis control automation [19], training [20], and testing and evaluation. The fourth application is inference of hidden change in context to detect fraud and computer network intrusion and to monitor web transactions.

The supervised learning task is represented as a discrete classification (concept learning) problem over continuous-valued input. It can be systematically decomposed by partitioning the input attributes (or fields) based on prior information such as *typing* of attributes (e.g., geographical, automobile-specific demographics, driver-specific demographics, etc.). State space search may be applied to automatically search for partitions even if no such information is available [16, 4], but this research focuses on how knowledge about *attribute relevance* may be exploited. Clustering, or vector quantization, is then performed on the *partitioned* training data – i.e., the training data is restricted to one *subset of channels* in the partition on each application of clustering algorithm. This produces new intermediate training targets [1] and defines new learning *subtasks* (mappings from a subset of the input channels to an intermediate target, or codebook, defined by clustering).

For these experiments, nearest-neighbor (Voronoi) tesselation, regression, and feedforward ANNs are used on the resultant learning subtasks, as depicted in Figure 3. An important contribution of this work is that the number of cluster centers is determined by setting a threshold on the number of exemplars (insurance policies, text documents, time series observations) that belong to a cluster. Thus, the specification of the number of cluster centers is made in terms of an independent criterion, plus the output of SOM, instead of by trial and error.

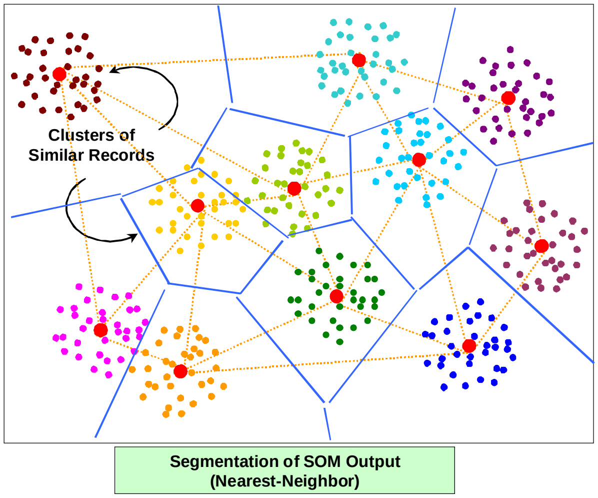


Figure 3. Output of cluster segmentation algorithm (labeled by a nearest-neighbor algorithm)

The solid lines connecting cluster centers denote the Delaunay triangulation, the dual of the Voronoi diagram (shown with solid lines). Quantization error is due to assumptions of linear regions that can be relaxed using higher order Voronoi diagrams or more regions. The mixture estimation task is completed using ANNs. The mixture model, comprising the expert (or “specialist”) and mixture estimation (or “moderator”) subnetworks, is referred to as a *specialist-moderator* network. An algorithm for its construction and training is presented in [16].

Data Preparation

The SOM-based architecture was tested on a classification problem over a large (350,000-record) database of automobile policy records from several U.S. states. The original input consisted of 471 attributes, from which 225 were selected using domain knowledge. The pre-filtered input was further partitioned according to demographic attribute type.

The clustering algorithm, applied to each subset of inputs, produced a task decomposition along equivalence classes of attribute types – yielding a vector quantization (i.e., codebook) appropriate for each subset of input. A similar method was used to group overall risk levels into *tiers* and the objectives, *total loss* (in dollars) and *loss ratio* (in dollars per normalized unit of premium), into *bins*. Two families of experiments were performed: one to classify individual policies, one to classify a random sample (of size 1000) by aggregate objective (sum of total loss or ratio of total loss). All supervised learning components were trained using error backpropagation with momentum.

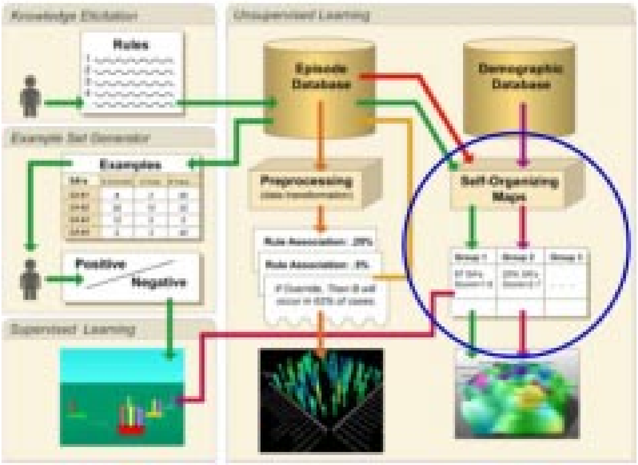


Figure 4. A framework for KDD-based anomaly (fraud and catastrophe) prediction

Figure 4 illustrates our framework for KDD-based anomaly detection using SOM as an unsupervised learning component of an interactive rule refinement system. This system entails elicitation of *prefiltering queries* to be made against an episodic database (historical database of transactions), which are used to identify anomalous or "interesting" transactions. These are labeled through inspection by a human expert, and are used to train an inducer [17] on clusters (intermediate concepts) found by applying SOM to both the original episodic data and additional demographic data.

Automatic Construction of Hierarchical Mixtures

Clustering, or vector quantization, is then performed on the *partitioned* training data – i.e., the training data is restricted to one *subset of channels* in the partition on each application of clustering algorithm. This produces new intermediate training targets and defines new learning *subtasks* (mappings from a subset of the input channels to an intermediate target, or codebook, defined by clustering). Preliminary experiments used two clustering algorithm: Kohonen's Self-Organizing Feature Map (SOFM or SOM) [12], and the simple algorithm described in [13]. For the multimodal sensor integration experiment, which involves time series data, simple recurrent networks (SRNs) of the Elman, Jordan, and *input recurrent* (exponential trace memory) [14, 13] variety are used on the resultant learning subtasks. The input recurrent variety was found to yield the highest mixture estimation accuracy on cross-validation data.

Applications and Experimental Results

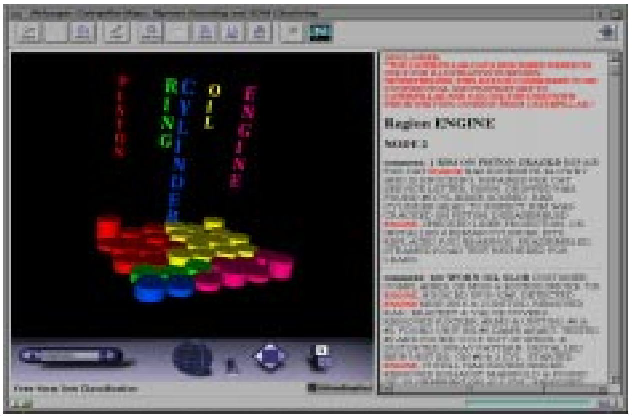
Automobile Insurance Risk Valuation

The new mixture model achieves higher classification accuracy than non-modular networks and Hierarchical Mixtures of Experts (HME) on this problem. It also requires fewer training parameters and converges more quickly than a stacked network of feedforward ANNs, while achieving equal classification accuracy with respect to actual financial loss.

Text Document Categorization

Figure 5 depicts the output of a SOM-based text mining system as applied to technical documents (repair reports filed by technicians). The training objective was to classify a set of reports by the *prevalent* (and *relevant*) keywords as extracted from the text and an accompanying corpus of *comments* (20-character summaries).

Current research attempts to boost classification accuracy using attribute partitioning and SOM-based decomposition. As the objective is to detect *emerging issues* (salient reliability issues in a product line), the problem lends itself naturally to decomposition.



Region ENGINE

This data is considered to be confidential and proprietary to Caterpillar and may only be used with prior written consent from Caterpillar.

Figure 5. Output of a SOM-based text mining system

Multisensor Integration in Crisis Monitoring

Our sensor fusion framework is part of a data reduction and synthesis system that comprises:

1. **Model Identification** – extraction of a data model in terms of alarm channels from on-board sensors (in ground vehicles and possibly avionic systems)
2. **Prediction Objective Specification** – the capability for the user to interactively define an analytical objective (e.g., prediction of a failure modes in reliability testing using high-volume data buses). This functionality provides decision support for tesing and evaluation objectives.

3. **Reduction** – simplification of the data model by selection and downsampling of data channels. Selection criteria are defined in terms of *relevance* to an analytical objective, such as online detection (prediction) of a hazard condition from time-indexed data.
4. **Synthesis** – the ability to generate new channels that improve prediction quality (i.e., reduce classification and localization error for hazard conditions).
5. **Integration -** of multiple (time series) models for nonlinear system identification based on selected, reduced, and synthesized data channels.

Preliminary experiments show that the efficiency of a time series analysis tool (in terms of relevance determination) can be boosted using prior knowledge and *wrappers* [6] for partition search [16].

Fraud/Intrusion Detection and Web Monitoring

Knowledge discovery from logs (records of user, client, server, vendor, or customer transactions) is a highly desired capability in business technology, as it has ramifications for electronic commerce and computer security.

Our framework for monitoring of sales transactions, depicted in Figure 4, extends to security applications when the model (e.g., a temporal ANN or HMM) is capable of capturing hidden changes in context and making probabilistic inferences (predictions) about the "next command" or "next transaction".

Conclusions

Novel Contributions

The novel theoretical contribution of this approach is its use of newly-formulated intermediate training targets, discovered using input partitioning and clustering, across mixture components. For applications exhibiting factorial structure, it provides a modular decomposition and discrete quantization (of intermediate training targets) that boosts classification accuracy. For example, in the automobile insurance policy classification test bed, this allows tiers (risk categories) to be reorganized to better predict loss

Current and Future Work

An important topic that this research continues to investigate is the process of automating task decomposition for model selection. We hypothesize that higher generalization quality can be achieved by broadening the repertoire of ANN classifiers. [16] describes a metric-based model selection system that is applied after cluster definition, to identify the "right tool" for each newly reformulated learning task. A key question that this line of research addresses is: how does the synthesis of attributes (as a method of task *decomposition*) support *relevance determination* [6] in a modular learning architecture?

Acknowledgements

The authors thank Tom Redman and John Martirano for implementing the distributed infrastructure of the KDD system described in this paper, Lisa Gatzke for preparing Figure 2, and Nina Mishra and Jason Hibbeler for developing the VRML-based viewer and SOM-based text mining system depicted in Figure 3.

References

[1] L.-M. Fu and B. G. Buchanan. Learning Intermediate Concepts in Constructing a Hierarchical Knowledge Base. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-85)*, pages 659-666, Los Angeles, CA, 1985.

[2] D. H. Wolpert. Stacked Generalization. *Neural Networks*, 5:241-259, 1992.

[3] D. P. Benjamin, editor. *Change of Representation and Inductive Bias*. Kluwer Academic Publishers, Boston, 1990.

[4] W. H. Hsu, S. R. Ray, and D. C. Wilkins. A Multistrategy Approach to Classifier Learning from Time Series. *Machine Learning, Special Issue on Multistrategy Learning*, to appear.

[5] K. Kira and L. A. Rendell. The Feature Selection Problem: Traditional Methods and a New Algorithm. In *Proceedings of the National Conference on Artificial Intelligence (AAAI-92)*, p. 129-134, San Jose, CA. MIT Press, Cambridge, MA, 1992.

[6] R. Kohavi and G. H. John. Wrappers for Feature Subset Selection. *Artificial Intelligence, Special Issue on Relevance, 97*(1-2):273-324, 1997.

[7] S. K. Donoho and L. A. Rendell. Rerepresenting and restructuring domain theories: A constructive induction approach. *Journal of Artificial Intelligence Research*, 2:411-446, 1995.

[8] L. Breiman. Bagging Predictors. *Machine Learning*, 24:123-140, 1996.

[9] T. Freund and R. E. Schapire. Experiments with a New Boosting Algorithm. In *Proceedings of ICML-96*.

[10] M. I. Jordan and R. A. Jacobs. Hierarchical Mixtures of Experts and the EM Algorithm. *Neural Computation*, 6:181-214, 1994.

[11] R. E. Stepp III and R. S. Michalski. Conceptual Clustering: Inventing Goal-Oriented Classifications of Structured Objects. In *Machine Learning: An Artificial Intelligence Approach*, R. S. Michalski, J. G. Carbonell, and T. M. Mitchell, editors. Morgan-Kaufmann, San Mateo, CA, 1986.

[12] T. Kohonen. The Self-Organizing Map. *Proceedings of the IEEE*, 78:1464-1480, 1990.

[13] S. R. Ray and W. H. Hsu. Self-Organized-Expert Modular Network for Classification of Spatiotemporal Sequences. *Journal of Intelligent Data Analysis*, 2(4), URL: http://www-east.elsevier.com/ida/browse/0204/ida00039/ida00039.htm. October, 1998.

[14] M. C. Mozer. Neural Net Architectures for Temporal Sequence Processing. In *Time Series Prediction: Forecasting the Future and Understanding the Past (Santa Fe Institute Studies in the Sciences of Complexity XV)*, A. S. Weigend and N. A. Gershenfeld, editors. Addison-Wesley, Reading, MA, 1994.

[15] T. Li, L. Fang, and K. Q-Q. Li. Hierarchical Classification and Vector Quantization With Neural Trees. *Neurocomputing* 5:119-139, 1993.

[16] W. H. Hsu. *Time Series Learning With Probabilistic Network Composites*. Ph.D. thesis, UIUC. URL: http://www.ncsa.uiuc.edu/People/bhsu/thesis.html, August, 1998.

[17] R. Kohavi, D. Sommerfield, and J. Dougherty. Data Mining Using *MLC++*: A Machine Learning Library in C++. In *Tools with Artificial Intelligence*, p. 234-245, IEEE Computer Society Press, Rockville, MD, 1996.

[18] R. Kohavi *et al*. *MineSet 2.5*. Silicon Graphics, Mountain View, CA, 1998.

[19] W. H. Hsu, N. D. Gettings, V. E. Lease, Y. Pan, and D. C. Wilkins. A New Approach to Multistrategy Learning from Heterogeneous Time Series. In *Proceedings of the International Workshop on Multistrategy Learning*, Milan, Italy. June, 1998.

[20] E. Grois, W. H. Hsu, D. C. Wilkins, and M. Voloshin. Bayesian Network Models for Automatic Generation of Crisis Management Training Scenarios. In *Proceedings of the National Conference on Innovative Applications of Artificial Intelligence (IAAI-98)*, pp. 1113-1120. Madison, WI. July, 1998.