

# FINDING STRUCTURE IN SIGNALS, IMAGES, AND DATA

*Erkki Oja*

Helsinki University of Technology  
Neural Networks Research Centre  
P.O. Box 5400, FIN-02015 HUT, Finland  
erkki.oja@hut.fi

## ABSTRACT

The talk will be a tutorial survey, concentrating on the main principles and categories of unsupervised neural learning in the problem of data mining for signals, images, and data. In neural computation, there are two classical categories for unsupervised learning methods and models: first, extensions of Principal Component Analysis and Factor Analysis, and second, learning vector coding or clustering methods that are based on competitive learning. The talk concentrates on two of these extensions: for the first category, the novel technique of Independent Component Analysis, and for the second category, the Kohonen Self-Organizing Map. The more recent trend in unsupervised learning is to consider this problem in the framework of probabilistic generative models. If it is possible to build and estimate a model that explains the data in terms of some latent variables, key insights may be obtained into the true nature and structure of the data. This approach is also briefly reviewed. After a brief introduction to the underlying theoretical foundations of these ideas, unsupervised neural learning will be illustrated by several applications in data mining ranging from document and pictorial databases to blind signal separation.

**Keywords** unsupervised learning, data mining, independent component analysis, self-organizing map

## Acknowledgement

This work was supported by the Finnish Centre of Excellence Programme (2000-2005) of the Academy of Finland, project New information processing principles, 44886.

## 1. INTRODUCTION

Progress in computer and information sciences was for a long time restricted by the state-of-the-art of computer hardware and data networks. In recent years a new situation has been encountered: the worldwide proliferation of powerful computing services has caused an uncontrolled flood

of information in the Internet and other media. It therefore becomes increasingly important to develop fundamentally new information processing principles for making relevant knowledge accessible to the user and to present it in a comprehensible form. This means, for example, completely new explorative data analysis and data mining methods, combined with advanced graphics facilities.

Along with the explosive increase in available digital data, the computing power of modern hardware has been dramatically increased as well. With the increasing computing power, it has become possible to digitally process and classify huge masses of natural data, such as statistical information, images, speech, text, as well as other kinds of signals and measurements coming from very different sources. Such tasks occur in industry, remote sensing, medicine, finance, and natural sciences, to mention only a few main fields. For financial, medical, administrative, and other databases, one needs efficient tools for visualization, prediction, clustering, and profiling. In industrial problems, it is essential to build empirical data based models of complex systems in order to be able to predict, monitor, diagnose faults, and control the systems.

One of the central tools in data mining is unsupervised learning. This means a completely data driven approach in which the pertinent structure, in the form of patterns, clusters, or models, is automatically found from the data using advanced statistical and computational techniques. Some insight into the unsupervised learning problem can be inferred from cognitive science. It is obvious that many effective computing principles that we do not yet know in detail exist in the biological brain. For example, many hierarchical computing structures of the brain have still remained a mystery. On the other hand, the mathematical operations and expressions that we use for the description of known neural operations can be computed digitally with much higher accuracy and stability than what is possible by the analog computing principles of the biological networks. Therefore, trying to combine the best of these two worlds is a strong motivation, emerging in the research field of neural computation. This can be seen as being situated at the intersection

of machine learning, computation, and advanced statistics.

The Section 2 of this paper reviews the three main approaches to unsupervised machine learning in neural networks. Then, Section 3 illustrates these approaches by some well-known concrete mathematical models. Section 4 mentions some applications that will be covered in detail in the talk.

This paper is based on the more extensive review (Oja, 2001).

## 2. WHAT IS UNSUPERVISED LEARNING

Unsupervised learning is a deep concept that can be approached from very different perspectives, from psychology and cognitive science to engineering. It is often called "learning without a teacher". This implies that a learning human, animal, or artificial system observes its surroundings and, based on these observations, adapts its behavior without being told how to associate given observations to given desired responses (supervised learning) or without even given any hints about the goodness of a given response (reinforcement learning). Usually, the result of unsupervised learning is a new explanation or representation of the observation data, which will then lead to improved future responses or decisions (Hinton and Sejnowski, 1999). This is precisely the problem in data mining, too.

In machine learning and artificial intelligence, such a representation is a set of concepts and rules between these concepts, which give a symbolic explanation for the data. In advanced statistics, the representation may be a clustering of the data, a discrete map, or a continuous lower-dimensional manifold in the vector space of observations, which explains their structure and may reveal their underlying causes.

Unsupervised learning seems to be the basic mechanism for sensory adaptation e.g. in the visual pathway (Barlow, 1989). If we accept the hypothesis that biological learning is based on synaptic modification, a big problem is how supervised learning rules like back-propagation could be implemented locally on the synaptic level. The biological substrate seems to be much more compatible with the unsupervised mode of learning. For more biologically oriented neural approaches, see (Grossberg, 1988). On the engineering side, unsupervised learning is a highly powerful and promising approach to some practical data processing problems like data mining and knowledge discovery from very large databases, or new modes of human-computer interactions in which the software adapts to the requirements and habits of the human user by observing her behaviour.

## 3. EXAMPLES OF UNSUPERVISED LEARNING IN NEURAL COMPUTATION

In neural computation, there have been two classical categories for unsupervised learning methods and models: first, extensions of Principal Component Analysis and Factor Analysis, and second, learning vector coding or clustering methods that are based on competitive learning (Haykin, 1999). The more recent trend in unsupervised machine learning is to consider this problem in the framework of probabilistic generative models (Hinton and Sejnowski, 1999). If it is possible to build and estimate a model that explains the data in terms of some latent variables, key insights may be obtained into the true nature and structure of the data. Operations like prediction and compression become easier and rigorously justifiable.

### 3.1. The Self-Organizing Map

The goal of unsupervised learning, finding a new compressed representation for the observations, can be interpreted as coding of the data. Thus learning vector coding methods that are based on competitive learning can be highly useful. A typical application is data mining or profiling from massive databases. It is of interest to find out what kind of typical clusters there are among the data records. In a customer profiling application, finding the clusters from a large customer database means more sharply targeted marketing with less cost. In process modelling, finding the relevant clusters of the process state vector in real operation helps in diagnosis and control. A competitive learning neural network gives an efficient solution to this problem. The best-known competitive learning network is the Self-Organizing Map (SOM) introduced by Kohonen (see Kohonen, 2001).

In vector coding, the problem is to place a fixed number of vectors, called *codewords*, into the input space which is usually a high-dimensional vector space. The input data (observations) are given as a training set of numerical vectors  $\mathbf{x}(1), \dots, \mathbf{x}(T)$ . For example, the inputs can be grayscale windows from a digital image, measurements from a machine or an industrial process, financial data describing a company or a customer, or pieces of English text represented by word histograms. The dimension  $n$  of the data vectors is determined by the problem and can be very large. In the WEBSOM system for organizing collections of text documents (Kohonen *et al.*, 2000), the dimensionality of the data in the largest applications is about  $n = 50,000$  and the size of the training sample is about  $T = 7,000,000$ .

The goal of SOM learning is not only to find the most representative code vectors for the input training set in the sense of minimum distance, as is the case in the usual vector coding methods, but at the same time to form a topological mapping from the input space to the grid or lattice of neurons. This idea originally stems from the modelling

of the topographic maps on the sensory cortical areas of the brain. A related early work in neural modelling is (Malsburg, 1973).

For any data point  $\mathbf{x}$  in the input space, one or several of the codewords are closest to it. Assume that  $\mathbf{w}_i$  is the closest among all codewords:

$$\|\mathbf{x} - \mathbf{w}_i\| = \min\|\mathbf{x} - \mathbf{w}_j\|, j = 1, \dots, k \quad (1)$$

The unit  $i$  having the weight vector  $\mathbf{w}_i$  is then called the *best-matching unit* (BMU) for vector  $\mathbf{x}$ . Note that for fixed  $\mathbf{x}$ , Eq. (1) defines the index  $i = i(\mathbf{x})$  of the BMU, and for fixed  $i$ , Eq. (1) defines the set of points  $\mathbf{x}$  that are mapped to that index and thus all belong to the same cluster. By the above relation, the input vectors  $\mathbf{x}$  are mapped to the discrete set of indices  $i$ .

By a topological mapping the following property is meant: if a given point  $\mathbf{x}$  is mapped to unit  $i$ , then all points in neighborhoods of  $\mathbf{x}$  are mapped either to  $i$  itself or to one of the units in the neighborhood of  $i$  in the lattice. Because no topological maps between two spaces of different dimensions can exist in the strict mathematical sense, a two-dimensional neural layer can only follow locally two dimensions of the multidimensional input space. Usually the input space has a much higher dimension, but the data cloud  $\mathbf{x}(1), \dots, \mathbf{x}(T)$  used in training may be roughly concentrated on a lower-dimensional manifold that the map is able to follow at least approximately (Kohonen, 2001). The well-known Kohonen learning rule is able to tune the map so that weight vectors attain optimal positions. For recent advances on the SOM, see (Oja and Kaski, 1999).

### 3.2. PCA, ICA, and FA

The other class of unsupervised learning methods is motivated by standard statistical methods like Principal Component Analysis (PCA) or Factor Analysis (FA), which give a reduced subset of linear combinations of the original input variables. A classical approach are the on-line PCA learning rules introduced by the author (Oja, 1982). As an example, consider here Factor Analysis (see e.g. Harman, 1967).

In FA, a generative latent variable model is assumed for the observation vectors  $\mathbf{x}$ :

$$\mathbf{x} = \mathbf{A}\mathbf{y} + \mathbf{n}. \quad (2)$$

FA was originally developed in social sciences and psychology. In these disciplines, the researchers want to find relevant and meaningful factors that explain observed results. The interpretation in the model (2) is that the elements of  $\mathbf{y}$  are the *unobservable* factors. The elements  $a_{ij}$  of the unknown matrix  $\mathbf{A}$  are called *factor loadings*. The elements of the unknown additive term  $\mathbf{n}$  are called specific factors. The elements of  $\mathbf{y}$  (the factors) are uncorrelated, zero mean and

gaussian, and their variances are absorbed into the matrix  $\mathbf{A}$  so that we may assume

$$E\{\mathbf{y}\mathbf{y}^T\} = \mathbf{I}. \quad (3)$$

The elements of vector  $\mathbf{n}$  are zero mean, uncorrelated with each other and also with the factors  $y_i$ ; denote  $\mathbf{Q} = E\{\mathbf{n}\mathbf{n}^T\}$ . It is a diagonal matrix. We may write the covariance matrix of the observations from (2) as

$$E\{\mathbf{x}\mathbf{x}^T\} = \mathbf{C}_x = \mathbf{A}\mathbf{A}^T + \mathbf{Q}. \quad (4)$$

In practice, we have a good estimate of  $\mathbf{C}_x$  available, given by the sample covariance matrix. The main problem is then to solve the matrix  $\mathbf{A}$  of factor loadings and the diagonal covariance matrix  $\mathbf{Q}$  such that they will explain the observed covariances from (4). There is no closed-form analytic solution for  $\mathbf{A}$  and  $\mathbf{Q}$ . Assuming  $\mathbf{Q}$  is known or can be estimated, we can solve  $\mathbf{A}$  from  $\mathbf{A}\mathbf{A}^T = \mathbf{C}_x - \mathbf{Q}$ . This solution is not unique, however: any matrix  $\mathbf{A}' = \mathbf{A}\mathbf{T}$  where  $\mathbf{T}$  is an orthogonal matrix ( $\mathbf{T}\mathbf{T}^T = \mathbf{I}$ ) will also be a solution. Then the factors will change to  $\mathbf{y}' = \mathbf{T}^T\mathbf{y}$ . For  $\mathbf{A}'$  and  $\mathbf{y}'$ , the FA model (2) holds, and the elements of  $\mathbf{y}'$  are still uncorrelated. The reason is that the property of uncorrelatedness is invariant to orthogonal transformations (rotations). Note that because the factors are uncorrelated and gaussian, they are also independent.

In *Independent Component Analysis* (ICA) (see e.g. Amari, 1996; Bell and Sejnowski, 1995; Cardoso, 1998; Hyvärinen, Karhunen and Oja, 2001; Jutten, 1991), the same model (2) is assumed, but now the assumption on  $y_i$  is much stronger: we require that they are *statistically independent* and *non-gaussian*. Interestingly, then the ambiguity in Factor Analysis disappears and the solution, if we can find one, is (almost) unique.

In the simplest form of ICA, the additive noise  $\mathbf{n}$  is not included and the standard notation for the independent components or *sources* is  $s_i$ ; thus the ICA model for observation vectors  $\mathbf{x}$  is

$$\mathbf{x} = \mathbf{A}\mathbf{s}. \quad (5)$$

It is again assumed that both  $\mathbf{x}$  and  $\mathbf{s}$  are zero mean. The observations  $x_i$  are now linear combinations or mixtures of the sources  $s_j$ . The matrix  $\mathbf{A}$  is called in ICA the *mixing matrix*. In a typical application of ICA, a set of parallel time signals such as speech waveforms, electromagnetic measurements from the brain, or financial time series, are assumed to be linear combinations of underlying independent latent variables. The variables, which are now the independent components, are found by efficient ICA learning rules.

A recent survey on ICA is (Hyvärinen, Karhunen and Oja, 2001) that also contains an extensive list of citations to the original literature.

### 3.3. Nonlinear Generative Models

The concept of a generative model is very general and potentially powerful. In fact, as discussed by (Roweis and Ghahramani, 1999), a large number of central techniques like FA, PCA, ICA, mixtures of Gaussians, vector quantization, and also dynamical models like Kalman filters or Hidden Markov Models, can be presented in a unified framework of unsupervised learning under a single basic generative model. In the Bayesian Ying - Yang model (Xu 2000), likewise a generic framework of unsupervised learning is employed for the basic data models, both static and temporal.

We already saw examples of generative models in the case of Factor Analysis and Independent Component Analysis. Also Principal Component Analysis can be derived from a generative model in the technique called Probabilistic PCA (Tipping and Bishop, 1999). A problem with such linear models, however, is that they cannot represent well data that is not a linear mixture of some underlying gaussian or nongaussian variables. For data clouds that have an irregular or curved shape, these methods fail.

In the Generative Topographic Map (GTM) algorithm (Bishop *et al.*, 1998), the observation vectors  $\mathbf{x}$  are expressed in terms of a number of latent variables, which are defined on a similar lattice or grid as the neurons in the SOM. The mapping from the latent variables to the observations is *non-linear*:

$$\mathbf{x} = \mathbf{f}(\mathbf{y}, \mathbf{M}) + \mathbf{n} \quad (6)$$

where  $\mathbf{M}$  is an array of parameters of the nonlinear function  $\mathbf{f}$ , and  $\mathbf{n}$  is additive noise. The form of the function  $\mathbf{f}$  is assumed to be determined except for the unknown parameters. The model (6) is the generative latent variable model of the GTM method. It means that the observed data vectors  $\mathbf{x}$  are basically concentrated on a lower dimensional nonlinear manifold in the data space, except for the additive noise. The vectors  $\mathbf{w}_i = \mathbf{f}(\mathbf{y}_i, \mathbf{M})$  that are the images of the node points  $\mathbf{y}_i$  are analogous to the weight vectors or codewords of the SOM. If  $\mathbf{f}$  is smooth, a topographic ordering for the codewords is automatically guaranteed, if such an ordering is valid for the latent points  $\mathbf{y}_i$ . The GTM also has the advantage that it postulates a smooth manifold that naturally interpolates between the code vectors  $\mathbf{w}_i$ . The parameters can be learned using the EM algorithm.

When comparing the FA model (2) and the GTM model (6), certain similarities emerge: both have a number of latent variables, given by the vector  $\mathbf{y}$ , and additive gaussian noise  $\mathbf{n}$ . In FA, the mapping from  $\mathbf{y}$  to the data  $\mathbf{x}$  is linear, in GTM it is nonlinear. Another clear difference is that in FA, the factors are gaussian, while in GTM, the prior density  $p(\mathbf{y})$  for the latent factors has a very special (atomic) form.

Another possibility for this density in the nonlinear case,

too, would be the gaussian density, which would then be close to the original flavor of FA. If we assume that the prior for  $\mathbf{y}$  is gaussian with unit (or diagonal) covariance, making the elements  $y_i$  independent, as in eq. (3), then the model (6) may be called *nonlinear factor analysis*. A further extension would be  $p(\mathbf{y})$  that is *nongaussian but factorizable* so that the  $y_i$  are independent; then the model becomes *non-linear independent component analysis*.

Recently, (Valpola, 2000) used an approximation for the nonlinear function  $\mathbf{f}(\mathbf{y}, \mathbf{M})$  in the model, that was based on a Multilayer Perceptron (MLP) network with one hidden layer. It is well-known (see e.g. Haykin, 1998) that this function can approximate uniformly any continuous functions on compact input domains and it is therefore suitable for this task. Then the model becomes

$$\mathbf{x} = \mathbf{B}\phi(\mathbf{A}\mathbf{y} + \mathbf{a}) + \mathbf{b} + \mathbf{n} \quad (7)$$

where  $\mathbf{A}$ ,  $\mathbf{a}$  are the weight matrix and offset vector of the hidden layer,  $\phi$  is the sigmoidal nonlinearity, typically a tanh or  $\sinh^{-1}$  function, and  $\mathbf{B}$ ,  $\mathbf{b}$  are the weight matrix and offset vector of the linear output layer. It is understood that  $\phi$  is applied to its argument vector element by element. In practice, there is a training sample  $\mathbf{x}(1), \dots, \mathbf{x}(T)$ , and we wish to solve from the model the corresponding source or factor vectors  $\mathbf{y}(1), \dots, \mathbf{y}(T)$ .

The problem now is that, contrary to the usual supervised learning situations, the inputs to the MLP are not known and therefore back-propagation type of learning rules cannot be used for finding the unknown parameters. The idea in (Valpola, 2000) is to use a purely Bayesian approach called *ensemble learning*. The cost function is the Kullback - Leibler divergence between the true posterior probability for the parameters, given the observations, and an approximation of that density. Several applications with real data have been shown. The model has also been extended to a dynamical model, similar to an extended Kalman filter but with unknown parameters, and very promising results are obtained in case studies.

## 4. APPLICATIONS

The talk will be an introductory survey, concentrating on the main principles and categories of unsupervised learning. In the talk, the theoretical foundations of unsupervised machine learning will be shortly reviewed and the techniques will be illustrated by several applications in data mining: finding relevant documents in large document collections, content-based image retrieval, finding structure in biomedical measurements, and finding hidden nonlinear factors in time series. For more information and references, see the Web pages (NNRC, 2001).

## References

- Amari, S.- I., Cichocki, A. and Yang, H., "A new learning algorithm for blind source separation". In *Advances in Neural Information Processing Systems 8*, Cambridge: MIT Press, 1996, pp. 757 - 763.
- Barlow, H. (1989). Unsupervised learning. *Neural Computation 1*, 295-311.
- Bell, A. and Sejnowski, T., "An information-maximization approach to blind separation and blind deconvolution", *Neural Computation 7*, 1995, pp. 1129 - 1159.
- Bishop, C., Svensen, M and Williams, C., "GTM: the generative topographic mapping", *Neural Computation 10*, 1998, pp. 215 - 234.
- Cardoso, J.- F., "Blind signal separation: statistical principles", *Proc. of the IEEE 9 (10)*, 1998, pp. 2009 - 2025.
- Grossberg, S, *Neural networks and natural intelligence*. Cambridge, MA: MIT Press, 1988.
- Harman, H.H., *Modern Factor Analysis*. Univ. of Chicago Press, 1967.
- Haykin, S., *Neural Networks - a Comprehensive Foundation*. New York: MacMillan College Publ. Co., 1998.
- Hinton, G. and Sejnowski, T. (2000). *Unsupervised Learning - Foundations of Neural Computation*. MIT Press, Cambridge.
- Hyvärinen, A., Karhunen, J., and Oja, E. (2001). *Independent Component Analysis*. Wiley-Interscience, New York.
- Jutten, C. and Herault, J., "Blind separation of sources, part I: an adaptive algorithm based on neuromimetic architecture", *Signal Processing 24*, 1991, pp. 1 - 10.
- Kohonen, T. (2001). *The Self-Organizing Map*. Springer, Berlin.
- Kohonen, T., Kaski, S., Lagus, K., Salojärvi, J., Paatero, V. and saarela, A., "Self organization of massive document collection", *IEEE Trans. Neural Networks 11 (3)*, 2000, pp. 574 - 585.
- von der Malsburg, C., "Self-organization of orientation sensitive cells in the striate cortex", *Kybernetik 14*, 1973, pp. 85 - 100.
- NNRC (2001). Web pages of the Neural Networks Research Centre, Helsinki University of Technology. [Online reference, see <<http://www.cis.hut.fi/research/>>, referred 20th July, 2001].
- Oja, E., "A Simplified Neuron Model as a Principal Components Analyzer", *J. Math. Biol. 15*, 1982, pp. 267-273.
- Oja, E., "Unsupervised learning in neural computation". *Theor. Comp. Science*, to appear (2001).
- Oja, E. and Kaski, S. (Eds.), *Kohonen Maps*. Amsterdam: Elsevier, 1999.
- Roweis, S. and Ghahramani, Z., "A unifying review of linear gaussian models", *Neural Computation 11 (2)*, 1999, pp. 305 - 346.
- Tipping, M. E. and Bishop, C. M., "Mixtures of probabilistic principal component analyzers", *Neural Computation 11 (2)*, 1999, pp. 443 - 482.
- Xu, L., "Temporal BYY learning for state space approach, hidden Markov model, and blind source separation", *IEEE Trans. Signal Proc. 48*, 2000, pp. 2132 - 2144.