Unsupervised Learning With Global Objective Functions

Suzanna Becker
Department of Psychology
McMaster University
1280 Main Street West
Hamilton, Ontario, L8S 4K1
Canada

and

Richard S. Zemel
Department of Computer Science
University of Toronto
6 King's College Rd.
Toronto, Ontario, M5S 3H5
Canada

RUNNING HEAD: Unsupervised learning

Correspondence:
Suzanna Becker
Department of Psychology, McMaster University
1280 Main Street West, Hamilton, Ontario, Canada L8S 4K1
Phone: (905) 525-9140 ext. 23020
Fax: (905) 529-6225
email: becker@mcmaster.ca

INTRODUCTION

Unsupervised learning algorithms can be distinguished by the absence of any supervisory feedback from the external environment. Often, however, there is an implicit *internally-derived* training signal. This training signal is based on some measure of the quality of the network's internal representation. The main problem in unsupervised learning research is to formulate a performance measure or cost function for the learning, to generate this internal supervisory signal. The cost function is also known as an objective function, since it sets the objective for the learning process. In this article, we review the most promising algorithms for unsupervised learning. We particularly focus on two types of learning procedures: those based on information-theoretic performance measures, and those employing maximum-likelihood density estimation. Another important class of biologically motivated learning algorithms, based on the idea of reinforcement – whether it be externally provided or internally generated – is covered in the articles "Reinforcement Learning" and "Hierarchical Reinforcement Learning".

Global objective functions or synaptic learning rules?

Since our concern is with unsupervised learning in *networks* and their global behaviour, we will focus on algorithms based upon globally-defined objective functions, rather than synaptic learning rules. By viewing the learning process as the optimization of a global objective

function, we can reduce a global algorithm into synaptic-level steps (weight changes), but the converse is not necessarily true; i.e., a given synaptic learning rule may not correspond to the derivative of any global objective function. Further, a well-defined objective function for the learning allows us to make global predictions about its behavior which are typically not possible in a bottom-up approach. Finally, the global objective function provides a quantitative measure of the success, or at least convergence, of the learning procedure.

In contrast to this top-down approach, many computational models of learning have been based on synaptic or cellular constraints, such as Hebb's postulate, and more recently, conditions for LTP induction. Hebb postulated that a synapse's efficacy should increase whenever the pre- and post-synaptic neurons are co-active. Hebb's postulate has gained popularity among neurobiologists as a plausible candidate for a cortical synaptic learning mechanism. A typical instantiation of Hebb's rule relates the evolution of the synaptic weight, w_{ij}, to the product of the pre- and post-synaptic activities, y_i, and y_j, as follows: $\Delta\ w_{ij} = \varepsilon\ y_i\ y_j$, where ε is a learning rate constant. While this rule and its variants (see POST-HEBBIAN RULES) provide a useful way to model cellular- and synaptic-level phenomena, they do not give us much insight into systems-level phenomena arising from neural plasticity. A large multi-layered network of neurons all following the same Hebbian rule does not generate particularly useful pattern processing abilities. Each neuron would tend to behave in a greedy fashion. If we hope to understand large-scale networks of the

brain, such as the visual system, we must find more global or network-level constraints on the learning, such as predicting the sensory input over time, that would cause the entire network of neurons to evolve cooperatively toward this common goal. By the same token, once a systems-level goal for the learning has been identified and synaptic-level weight updates have been derived, it is of interest to computational modellers to try to translate their global learning procedures into local, biologically plausible learning rules such as Hebbian learning.

Self-organization in perceptual systems

One of the major motivations for studying unsupervised learning is to discover the general computational principles underlying brain self-organization. Evidence of experience-dependent plasticity has been reported in a wide variety of brain areas. Perhaps the most startling evidence comes from a series of studies by Sur and colleagues (reviewed in Sur, 1989), who found that by artificially rerouting primary visual cortical input pathways to the auditory cortex in ferrets, the "auditory" cortical cells develop responses to visual stimuli, and exhibit characteristics of typical visual cortical receptive fields. These and other experiments have led to the characterization of a plastic brain capable of dynamical restructuring. Thus, the goal of biologically motivated learning research may be stated as the search for the objective function(s) employed by the brain.

INFORMATION-PRESERVING ALGORITHMS

Since there is no external teaching signal for unsupervised learning, the goal of the learning must be stated solely in terms of some transformation on the input which will preserve the interesting structure. The first task then is to define what constitutes interesting structure. Perhaps the simplest possible goal is to try to preserve *all* of the information, for example, by simply memorizing the input patterns. Pattern-associators (see ASSOCIATIVE NETWORKS) operating in auto-associative mode can be used as such by storing each input pattern associated with itself. However, models that perform exact memorization tend to have very poor capacity, and are unable to generalize their knowledge to new inputs.

Minimizing reconstruction error

Given the limited ability of networks to store a set of patterns exactly, a better strategy might be to try to find a *compressed* representation of the patterns. This may be helpful for preprocessing noisy data, and for modelling early stages of perceptual processing. Later stages of processing may impose additional constraints on the data reduction process, such as the need to map inputs to actions and their consequences. A standard data compression technique is principal components analysis (PCA) (see PRINCIPAL COMPONENTS ANALYSIS). Several learning procedures (reviewed in Becker and Plumbley, 1996) have been developed which converge to the first N principal directions of the input distribution.

These methods are optimal with respect to minimizing the mean squared reconstruction error for linear networks. However, PCA will often fail to capture interesting structure such as clustering in the data.

A more general method for finding a compressed representation that minimizes reconstruction error is to use a nonlinear back-propagation network as an auto-encoder (Hinton, 1989), by making the desired states of the N output units identical to the states of the N input units on each case. Data compression can be achieved by making the number of hidden units $M < N$. Further, the features discovered by the hidden units may be useful for subsequent stages of processing such as classification. However, with complicated input patterns containing multiple features, it may not be possible to relate the activities of individual hidden units to specific features. One way to constrain the hidden unit representation is to add extra penalty terms to the objective function (see BACKPROPAGATION: BASICS AND NEW DEVELOPMENTS). For example, Zemel (1994) imposed a penalty term on hidden unit activations that caused these units to represent high-dimensional data as localized bumps of activity in a lower-dimensional constraint surface. This encouraged the hidden units to form a map-like representation that best characterizes the input. Other penalty terms lead to other forms of hidden representations, such as sparse, or combinatorial representations (see MINIMUM DESCRIPTION LENGTH APPLICATIONS OF NEURAL NETWORKS).

Direct minimization of information loss

Another approach to ensuring that the important information in the input is preserved in the output is to use concepts from information theory. This use of the term "information" is in a purely statistical sense, rather than the perhaps more intuitive notion that a sentence be more or less informative based on its semantic content. Viewing a neuron or neural network as a communication channel, one can calculate the rate of information loss through the channel. Atick (1992) provides an excellent review of this and related approaches, as well as a good introduction to information theory. Several learning procedures have been proposed which minimize the information loss in a network, subject to processing constraints (reviewed in Becker and Plumbley, 1996). The common feature of these methods is the preservation of mutual information between the input vector $\mathbf{x}$ and output vector $\mathbf{y}$:

$$I_{x;y} \;=\; H(\mathbf{x}) \;+\; H(\mathbf{y}) \;-\; H(\mathbf{x},\mathbf{y}) \tag{1}$$

where $H(\mathbf{x}) = -\int_{\mathbf{x}} p(\mathbf{x}) \log p(\mathbf{x}) dx$ is the entropy of random variable x with probability distribution $p(x)$, and $H(\mathbf{x},\mathbf{y}) = -\int_{\mathbf{x},\mathbf{y}} p(\mathbf{x},\mathbf{y}) \log p(\mathbf{x},\mathbf{y}) d\mathbf{x}\; d\mathbf{y}$ is the entropy of the joint distribution of $\mathbf{x}$ and $\mathbf{y}$. The mutual information between two variables is highest when the variables have high entropies individually, but their joint distribution has low entropy. For example, a variable x with a highly peaked probability distribution, $p(x)$, has very low

entropy, $H(x)$; it is highly predictable a priori, and therefore a given observation of the value of x provides very little information. In contrast, a variable with a uniform distribution has maximum entropy; it is completely unpredictable a priori, therefore one gains maximal information by having observed its value. Thus, $I_{x;y}$ is high when $\mathbf{x}$ and $\mathbf{y}$ are difficult to predict *a priori*, but $\mathbf{x}$ becomes predictable after being told $\mathbf{y}$ (and vice versa).

If the network is free of processing noise and has enough units, its output layer can convey all the information contained in the input simply by copying the input. In 1988, Linsker (for a review, see Linsker, 1997) first proposed applying the "Infomax principle" in the presence of Gaussian processing noise at the output layer for linear networks. When the input distribution is Gaussian, the entropy greatly simplifies from the expected value of a log of a Gaussian, to a function of the log of the variance (or the log determinant of the covariance matrix, for a multivariate Gaussian) (see, e.g., Atick, 1992). Hence, the information is:

$$I = 0.5 \log \left(\frac{|\mathbf{Q}^{\mathbf{y}}|}{V(n)} \right)$$

where $|\mathbf{Q}^{\mathbf{y}}|$ is the determinant of the covariance matrix of the output vector $\mathbf{y}$ (the signal plus noise) and $V(n)$ is the noise variance. Maximizing this quantity results in a tradeoff between maximizing the variances of the outputs, and decorrelating them, depending on the noise level. For a single output unit, this is equivalent to maximizing the output variance of a unit, and leads to a simple Hebb-like learning rule. In Linsker's more recent work, he has

extended this idea to networks with multiple units, nonlinearities and sparse coding (cited in Linsker, 1997).

A related optimality criterion proposed by Barlow (1989) is to find a minimally *redundant* encoding of the sensory input vector into an n-element feature vector, which should facilitate subsequent learning. If the n features are statistically independent, then the formation of new associations with some event V (assuming the features are also approximately independent conditioned on V) only requires knowledge of the conditional probabilities of V given each feature y_i, rather than complete knowledge of the probabilities of events given each of the 2^m possible sensory inputs. Barlow proposes that one could achieve featural independence by finding a *minimum entropy encoding*: an invertible code which minimizes the sum of the feature entropies (see VISUAL CODING, REDUNDANCY, AND "FEATURE DETECTION").

A number of algorithms for Independent Components Analysis (ICA) (for a review, see Lee, Girolami, Bell and Sejnowski, 2000) instantiate Barlow's principle by direct maximization of entropy. For example, in Bell and Sejnowski's algorithm (reviewed in Lee et al., 2000), a one-layer network of units with sigmoidal activation functions is able to solve the blind source separation problem: given a linear combination of N independent sources such as a mixture of acoustic signals, find a transformation to a set of N statistically independent outputs. Although this algorithm is limited in its applicability to dimensionality-preserving

mappings of linear mixtures, it has been applied successfully in a number of domains including EEG analysis, and thus gained widespread interest in the signal processing community. More recently, Linsker (1997) has proposed a more biologically plausible version of ICA, also based on entropy maximization. It permits fewer than N output components and employs information locally available to each neural unit.

Our brains may engage in something like blind source separation, for example, when performing auditory streaming. However, unlike many ICA algorithms that are limited in finding at most N features in an N-dimensional input, the brain has many more neurons than sensory inputs, and employs a sparse, overcomplete representation. Olshausen and Field (1996) proposed an alternative instantiation of Barlow's principle of redundancy reduction that results in such representations. Rather than directly manipulating the entropy of the coding, they minimized the following energy function:

$$E = -\sum_{x,y}\left[I(x,y) - \sum_i a_i\phi_i(x,y)\right]^2 - \lambda\sum_i S(\frac{a_i}{\sigma}) \qquad (2)$$

where the leftmost term is the squared reconstruction error as a function of the image $I(x,y)$ and the weighted unit activations $\phi_i a_i$, and the rightmost term is a nonlinear function (e.g., a zero-mean Gaussian) of the unit activations chosen to enforce sparseness. The sparseness constraint favours a small number of feature detectors being active at a given moment, while

permitting some redundancy in the representation. The reconstruction term insures the preservation of as much information in the input as possible. Thus, when exposed to visual images, the model tends to form local, partially overlapping receptive fields at a variety of spatial scales closely resembling those seen in early stages of the mammalian visual system (see VISUAL CODING, REDUNDANCY AND FEATURE DETECTION).

Preserving information within extracted features

The methods discussed so far try to extract useful structure from data while assuming minimal prior knowledge, and are good for modelling early sensory processing. But can unsupervised learning be applied beyond these preprocessing stages, to extract higher order features and build more abstract representations? One approach is to make constraining assumptions about the structure of interest, and build these constraints into the network's architecture or objective function.

Spatio-temporal coherence is a ubiquitous feature of sensory signals. Becker and Hinton's Imax learning procedure (reviewed in Becker, 1996) discovers coherent properties of the input by maximizing the mutual information between the *outputs*, y_a and y_b, of network modules that receive input from different parts of the sensory input (e.g., different modalities, or different spatial or temporal samples). Note how this objective function differs from the Infomax principle; the latter tries to retain *all* of the information in the input by maximizing

the mutual information between inputs and outputs, whereas Imax tries to extract only those features common to two or more distinct parts of the input.

Under Gaussian assumptions about the signal and noise, Becker and Hinton simplified the mutual information I down to a log ratio of two variances, that of the signal plus noise and the noise, to derive the following objective function for the learning:

$$I = 0.5 \log \frac{V(y_a + y_b)}{V(y_a - y_b)}$$

This measure tells how much information the average of y_a and y_b conveys about the common underlying signal, i.e., the feature which is coherent across the two input samples. When applied to networks composed of multi-layer modules that receive input from adjacent, non-overlapping regions of the input, Imax discovered higher order image features (i.e., features not learnable by single-layer or linear networks) such as stereo disparity in random dot stereograms. This illustrates how one part of the brain might self-organize within the visual modality to learn spatially predictive features. The idea could also be applied to the outputs of two different modalities to learn about their common causes (for a review of related work see Becker, 1996). For example, when we see an object and hear a sound, the two are often correlated and probably help us to learn object categories.

DENSITY ESTIMATION TECHNIQUES

So far, we have focused on algorithms that try to manipulate the information preserved by the network by imposing various processing constraints or representational assumptions. An alternative approach is to model directly the probability distribution over the input patterns. Many unsupervised learning procedures can be viewed in this way. The general approach is to assume *a priori* a class of models which constrains the general form of the probability density function; then search for the particular model parameters defining the density function most likely to have generated the observed data. This approach of developing *generative models* of data can be cast as an unsupervised learning problem by treating the network weights as the model parameters θ, and the overall function computed by the network as being directly related to the density function. The goal is to find model parameters that maximize the log likelihood that the model generated the data, x:

$$\log(L) = \sum_x \log(p(\mathbf{x} \mid \theta)) \tag{3}$$

Many of the information-maximization algorithms described above can also be derived from this generative approach. For example, the reconstruction error in autoencoder learning can be derived as a data-likelihood term, and the additional penalty terms in the different algorithms as particular priors over the hidden states (Zemel, 1994). And the Bell-Sejnowski

ICA algorithm can be obtained by a particular choice of prior over the sources and a noise model over the input (Pearlmutter and Parra, 1997).

Mixture models and competitive learning

One convenient and popular choice of prior model is a mixture of Gaussians. This model performs a type of cluster analysis, and is the basis for deriving two more biologically plausible models that we will consider afterwards. The prior assumption in this case is that each data point was actually generated by one of n Gaussians having different means μ_i, variances ${\sigma_i}^2$, and prior probabilities π_i. Fixing the model parameters μ_i, σ_i, and π_i, we can compute the probability of a given data point $\mathbf{x}$ under a mixture-of-Gaussians model as follows:

$$p(\mathbf{x} \mid \{\mu_i\}, \{\sigma_i\}, \{\pi_i\}) = \sum_{i=1}^{n} \pi_i P_i(\mathbf{x}, \mu_i, \sigma_i) \tag{4}$$

where $P_i(\mathbf{x}, \mu_i, \sigma_i)$ is the probability of $\mathbf{x}$ under the ith Gaussian. Applying Bayes' rule, we can also compute the probability that any one of the Gaussians generated the data point $\mathbf{x}$:

$$p(i \mid \mathbf{x}, \{\mu_j\}, \{\sigma_j\} \{\pi_j\}) = \frac{\pi_i \; P_i(\mathbf{x}, \mu_i, \sigma_i)}{\sum_{j=1}^{n} \pi_j \; P_j(\mathbf{x}, \mu_j, \sigma_j)} \tag{5}$$

Given these probabilities, we can now use as a cost function the log-likelihood of the data given the model:

$$\log(L) = \sum_x \log(p(\mathbf{x} \mid \{\mu_i\}, \{\sigma_i\}, \{\pi_i\}))$$

By maximizing this function, we can approximate the true probability distribution of the data, given our prior model assumptions. Note that by taking the log of L, we obtain a cost function which is a sum of log probabilities, rather than a product of probabilities, for each input pattern. The model parameters can then be adapted by performing gradient ascent in $\log(L)$. The Expectation-Maximization (EM) algorithm alternatingly applies equation 4 (the Expectation step) and adapts the model parameters (the Maximization step) to converge on the maximum likelihood mixture model of the data.

Competitive learning (see FEATURE DISCOVERY BY COMPETITIVE LEARNING) procedures can be viewed as performing a discrete approximation to the density estimation algorithm described above, but can be implemented in a more biologically realistic neural circuit. The general idea is that units compete to respond (e.g., by a winner-take-all activation function or lateral inhibition), so that only the winning unit in each competitive cluster is active. The winning unit learns by moving its weight vector closer to the current input pattern. Hence, each unit minimizes the squared distance between its weight vector and the patterns nearest to it, as in standard k-means clustering. This version of competitive learning is closely related to fitting a mixture of Gaussians model with equal priors π_i and

equal fixed variances $\sigma_i{}^2$. Using the EM algorithm, every unit (not just the winner) moves its mean closer to the current input vector, in proportion to the probability that it's Gaussian model accounts for the current input (equation 5). Competitive learning approximates this step by making a binary decision as to which unit accounts for the input. Thus, the same learning rule applies, except that the proportional weighting is replaced by an all-or-none decision.

Nowlan (1990) proposed a Maximum Likelihood Competitive Learning (MLCL) model for neural networks. Rather than only allowing the winner to adapt, each unit adapts its weights for every input case, in proportion to how strongly it responds on a given case. This is an online version of the EM algorithm for Gaussian densities with equal priors, and adaptive means and variances. Nowlan found this method to be superior to traditional competitive learning models on several classification tasks. Becker (1999) extended MLCL to a network that computes the priors using spatiotemporal contextual cues, allowing hierarchical clustering of features based on common contexts. The architecture of Becker's model was motivated by the laminar and columnar organization seen throughout the neocortex. It was shown to learn local views of an object in the first layer, and to group nearby views together into more view-tolerant representations in the second layer.

Combinatorial representations

A major limitation of mixture models and competitive learning is that they employ a 1-of-n encoding, in which a single unit or feature is assumed to explain each datum. A *multiple causes* model is more appropriate when the most compact data description consists of several independent parameters (e.g., color, shape, size). Some pioneering connectionist work in this area was done by Neal (1992). Neal's multilayer "sigmoid belief networks" (SBNs) resemble stochastic Boltzmann machines (see BOLTZMANN MACHINES), but they are strictly feedforward. Output states are held fixed on training patterns selected from the environment, while the hidden unit states are freely but noisily updated. The weights are adjusted so as to increase the probability of the hidden units generating the training patterns. The network thereby learns to represent features in the hidden layer which explain correlations in the pattern set. Unfortunately, Monte Carlo sampling is a prohibitively time-consuming way to search for good hidden layer features. Saul, Jaakkola and Jordan (1996) proposed a way around this employing a mean field approximation for SBNs.

A major challenge in this area of research is to develop multiple cause models for multi-layered networks with top-down feedback. Perhaps the most noteworthy attempts in this direction are the Helmholtz Machine developed by Hinton et al. (see Dayan, "Helmholtz Machines and Sleep-Wake Learning") and Rao and Ballard's model (1997). In the Helmholtz machine, the bottom-up weights embody a "recognition model"; that is, they are used to

produce the most probable set of hidden states given the data. At the same time, the top-down weights constitute a "generative model"; that is, they produce a set of hidden states most likely to have generated the data. The "wake-sleep algorithm" maximizes the log likelihood of this model and results in a simple and elegant delta rule for updating either set of weights:

$$\Delta w_{kj} = \varepsilon s_k^\alpha (s_j^\alpha - p_j^\alpha) \tag{6}$$

where p_j^α is the target state for unit j on pattern α, and s_j^α is the corresponding network state, a stochastic sample based on the logistic function of the unit's net input. Target states for the generative weight updates are derived from top-down expectations based on samples using the recognition model, whereas for the recognition weights, the targets are derived by making bottom-up predictions based on samples from the generative model. The Helmholtz machine is restricted in training either the generative or recognition connections at a given time. In contrast, Rao and Ballard's model (1997) interleaves the training of these connections. This model is based upon the extended Kalman filter. At each level, representational nodes combine top-down predictions with bottom-up information to produce two sources of prediction error: 1) a predicted internal state at the next time instant which is sent to the preceding layer for predicting the bottom-up input, and 2) a prediction of the

top-down input which is sent to the subsequent layer. Training is cast within a maximum likelihood framework: the model fit depends upon the two sources of error mentioned above, as well as model cost terms for each of the model parameters. The model is presented as an account of learning and real-time processing in the visual cortex, and is shown to develop realistic local receptive fields as well as object-level representations when trained on natural image sequences.

DISCUSSION

We have argued in favor of the "global objective function" approach to modelling unsupervised learning processes, and explored several powerful learning procedures based on this approach. These methods have had success in modelling early perceptual processing. With the incorporation of highly constraining prior models, unsupervised learning procedures can form even more abstract representations of data, and extract higher-order features. A major direction of ongoing research is aimed at finding tractable instantiations of these learning procedures, and to apply them in multiple learning stages to form a diversity of representational levels. Recent work by Hinton and colleagues on tractable versions of the Boltzmann machine is a promising example of such efforts.

Acknowledgments

The authors acknowledge support from the Natural Sciences and Engineering Research Council of Canada to S.B. and the Office of Naval Research to R.Z. Many thanks also to Michael Arbib, Chris Williams and the anonymous reviewers for helpful comments.

References

Atick, J. J. (1992). Could information theory provide an ecological theory of sensory processing? Network, 3:213-251.

Barlow, H. B. (1989). Unsupervised learning. Neural Computation, 1:295-311.

Becker, S. (1996). Mutual information maximization: Models of cortical self-organization. Network: Computation in Neural Systems, 7:7-31.

Becker, S. (1999). Implicit learning in 3d object recognition: The importance of temporal context. Neural Computation, 11(2):347-374.

Becker, S. and Plumbley, M. (1996). Unsupervised neural network learning procedures for feature extraction and classification. International Journal of Applied Intelligence,6(3).

Hinton, G. E. (1989). Connectionist learning procedures. Artificial Intelligence, 40:185-234.

Lee, T., Girolami, M., Bell, A., and Sejnowski, T. (2000). A unifying information-theoretic framework for independent component analysis. Computers and Mathematics with Applications, 39:1-21.

Neal, R. M. (1992). Connectionist learning of belief networks. Artificial Intelligence, 56:71-113.

Nowlan, S. J. (1990). Maximum likelihood competitive learning. Neural Information Processing Systems, Vol. 2, (D.S. Touretzky, Ed.), San Mateo, CA: Morgan Kaufmann, pp. 574-582.

Olshausen, B. A. and Field, D. J. (1996). Emergence of simple-cell receptive field properties by learning a sparse code for natural images. Nature, 381:607-609.

Pearlmutter, B. A. and Parra, L. C. (1997). Maximum likelihood blind source separation: A context-sensitive generalization of ICA. Neural Information Processing Systems, Vol. 9 (M. Mozer, M. Jordan and T. Petsche, Eds.), MIT Press, pp. 613-619.

Rao, R. P. N. and Ballard, D. H. (1997). Dynamic model of visual recognition predicts neural response properties in the visual cortex. Neural Computation, 9:721-763.

Saul, L. K., Jaakkola, T., and Jordan, M. I. (1996). Mean field theory for sigmoid belief networks. Journal of Artificial Intelligence Research, 4:61-76.

Sur, M. (1989). Visual plasticity in the auditory pathway: Visual inputs induced into auditory thalamus and cortex illustrate principles of adaptive organization in sensory systems. In Arbib, M. and Amari, S., editors, *Dynamic Interactions in Neural Networks; Models and Data*, pages 35–51. Springer-Verlag.

Zemel, R. S. (1994). *A Minimum Description Length Framework for Unsupervised Learn-*

ing. PhD thesis, University of Toronto.