

# Variational Learning for Noisy-OR Component Analysis

Tomas Singliar and Milos Hauskrecht  
Department of Computer Science  
University of Pittsburgh  
Pittsburgh, PA 15260  
{*tomas, milos*}@cs.pitt.edu

## Abstract

Latent factor models offer a very useful framework for modeling dependencies in high-dimensional multivariate data. In this work we investigate a class of latent factor models with hidden noisy-or units that let us decouple high dimensional vectors of observable binary random variables using a 'small' number of hidden binary factors. Since the problem of learning of such models from data is intractable, we develop its variational approximation. We analyze special properties of the optimization problem, in particular its "built-in" regularization effect and discuss its importance for model recovery. We test the noisy-or model on an image deconvolution problem and illustrate the ability of the variational method to successfully learn the underlying image components. Finally, we apply the latent noisy-or model to analyze citations in a large collection of Statistical Machine Learning papers and show the benefit of the model and algorithms by discovering useful and semantically sound components characterizing the dataset.

**Keywords:** Learning, Variational methods, Bayesian networks.

## 1 Introduction

Latent variable models [14, 2] provide a very useful framework for modeling dependencies in high dimensional data. The models are often used in the component analysis where we want to identify characteristics of a small number of underlying components (factors, sources, or signals) that combine into the expression of observed high dimensional data. Examples of latent factor models include probabilistic principal component analysis [18, 3], mixtures of factor analysers [1], multinomial PCA (or aspect) models [5, 10, 4], multi-cause model [9, 16], or other independent component analysis frameworks [1, 15]. In addition to their role in modeling and understanding the structure of high-

dimensional data, latent factor models used in the component analysis can be applied in the dimensionality reduction where the vector of hidden factors represents a low-dimensional representation of the data sample.

In this work we investigate a special class of latent factor models that let us represent high-dimensional multivariate distributions of binary attributes and their local dependencies. The dependencies are modeled in terms of a small number of hidden binary factors that are combined through noisy-or units. Intuitively, noisy-or units let us model local dependencies (couplings) among observable components in the data indirectly – in terms of hidden factors and their values. Such a framework is especially useful if we want to model random binary variables with confounded stochastic fluctuations. However, it can be also applied in more general settings to approximate local dependencies among random variables. We believe that models with such characteristics can be very useful and applied to represent stochastic dependencies among components of large distributed systems, such as failures or congestions in transportation networks, spread of disease in epidemiology, and others.

The key step of component analysis corresponds to the learning of the parameters of the latent factor model from the data. Once the model is learned it can be used to make inferences on hidden factors, such as to identify the document topics in the aspect model [10, 4] or regulatory signals in the microarray DNA data [13]. In the statistical sense the learning corresponds to the estimation of parameters of the model. The limitation of latent factor models is the complexity of the learning problem; the standard EM formulation (decomposition) becomes exponential in the number of hidden factors. Variational approximations offer one possible solution to make the learning task more efficient, but at some loss of accuracy. To address this problem, we develop and test a variational learning algorithm for optimizing the parameters of the noisy-or network with hidden factors. Our algorithm builds upon and extends the work of

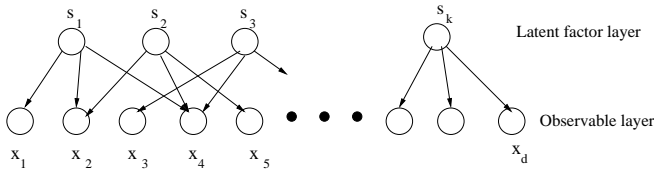


Figure 1: A bipartite belief network structure of the latent factor models with noisy-or units.  $\mathbf{x}$  is a vector of binary random variables that are observable and  $\mathbf{s}$  is a vector of hidden binary factors. Latent factors and noisy-or units model local interactions between components of  $x_j$ .

Jaakkola and Jordan who focused on and developed methods of variational inference for noisy-or networks [11]. The methods for learning a noisy-or network model with hidden components have not yet been investigated, to our knowledge. A very restricted model was explored by Kearns and Mansour [12] but their algorithm is exponential in the maximum number of hidden factors contributing to any observable variable. Our algorithm does not make any structural assumption and it is able to recover very well the active (nonzero) structure of a noisy-or network.

In the following text, we first describe the noisy-or network model with hidden units and its limitation in efficient inferences. Next we analyze the problem of learning the parameters of the latent factor network from data and point out the shortcomings of the exact Expectation-Maximization (EM) technique associated with its computational complexity. To alleviate the problems with the exact EM we develop and present its variational approximation. We test the model and the approximation algorithm on a synthetic image deconvolution problem. We investigate two aspects of the approach: recovery of complex multivariate distributions and dimensionality reduction, and show a very good performance of the algorithm on both of these tasks. Finally, we apply the model to analyze citations in a large collection of Machine Learning papers. We show the benefit of the new model and the algorithm by discovering useful and semantically sound subcommunities characterizing the dataset.

## 2 Latent Factor Model with Noisy-or Units

Consider a latent variable model with bipartite belief network structure illustrated in Figure 1. Nodes in the top layer represent a vector of latent factors  $\mathbf{s} = \{s_1, s_2, \dots, s_K\}$  with binary  $\{0, 1\}$  values and nodes in the bottom layer an observable vector of binary variables  $\mathbf{x} = \{x_1, x_2, \dots, x_d\}$ . We assume that  $\mathbf{x}$  is a

high-dimensional vector and that  $d > K$ . The connections between two layers of the bipartite graph represent dependencies among the components of the observable variables: the nodes coupled by one of the latent factor nodes are assumed to exhibit a local dependency pattern. The probabilistic dependency between nodes in the two layers is modeled via the noisy-or conditional distribution, which gives us a compact (low-complexity) parameterization of the relation among configurations of hidden factors and observable variables. The parameters  $\Theta$  of the model consist of:

- a set of prior probabilities  $\pi_i$  parameterizing the (Bernoulli) prior distributions  $P(s_i)$  for every hidden factor  $i$ ;
- a set of probabilities  $p_{ij}$  representing parameters of noisy-or conditional probability tables, one for each pair of hidden factor  $i$  and observed component  $j$ .

The structure of the model is similar to the QMR-DT model used for diagnosis in internal medicine [17]; the difference is that the top layer variables in our model are hidden. Their sole purpose is to model stochastic interaction patterns among observable variables in  $\mathbf{x}$ .

### 2.1 The joint distribution over observables

Given the bipartite model, the joint probability of an observation vector  $\mathbf{x}$  is defined as:

$$P(\mathbf{x}) = \sum_{\{\mathbf{s}\}} \left( \prod_{j=1}^d p(x_j|\mathbf{s}) \right) \left( \prod_{i=1}^K p(s_i) \right), \quad (2.1)$$

where  $\{\mathbf{s}\}$  denotes the sum over all configurations of  $\mathbf{s}$ , and  $P(s_i)$  is the prior probability of a hidden factor  $s_i$ . Given a vector of hidden binary factors  $\mathbf{s}$ , the conditional probability  $p(x_j|\mathbf{s})$  for an observable random component  $x_j \in \{0, 1\}$  is obtained through a noisy-or model as:

$$P(x_j|\mathbf{s}) = \left[ 1 - (1 - p_{0j}) \prod_{i=1}^K (1 - p_{ij})^{s_i} \right]^{x_j} \cdot \left[ (1 - p_{0j}) \prod_{i=1}^K (1 - p_{ij})^{s_i} \right]^{(1-x_j)} \quad (2.2)$$

where  $p_{0j}$  is the leak probability that models “all other” causes. The Equation 2.2 can be reparameterized with  $\theta_{ij} = -\log(1 - p_{ij})$  to obtain:

$$P(x_j|\mathbf{s}) = \exp \left[ x_j \log \left( 1 - \exp \left\{ -\theta_{0j} - \sum_{i=1}^K \theta_{ij} s_i \right\} \right) + (1 - x_j) \left( -\theta_{0j} - \sum_{i=1}^K \theta_{ij} s_i \right) \right]. \quad (2.3)$$

**2.2 Factorization** The bottleneck in computing the joint probability over observables  $P(\mathbf{x})$  in Equation 2.1 is the sum that ranges over all possible latent factor configurations, and thus, it is exponential in  $K$ . It is easy to see that if  $P(x_j|\mathbf{s})$  for both  $x_j = 0$  and  $x_j = 1$  can be expressed as:

$$P(x_j|\mathbf{s}) = \prod_{i=1}^K h(x_j|s_i), \text{ such that } \forall i, j : h(x_j|s_i) \geq 0 \quad (2.4)$$

then the full joint and the joint over the observables  $P(\mathbf{x})$  decompose as:

$$\begin{aligned} P(\mathbf{x}, \mathbf{s}) &= \prod_{j=1}^d P(x_j|\mathbf{s}) \prod_{i=1}^K P(s_i) = \\ &= \prod_{i=1}^K \left( P(s_i) \prod_{j=1}^d h(x_j|s_i) \right), \quad (2.5) \\ P(\mathbf{x}) &= \sum_{\{\mathbf{s}\}} \prod_{i=1}^K \left( P(s_i) \prod_{j=1}^d h(x_j|s_i) \right) \\ &= \prod_{i=1}^K \left( \sum_{\{s_i\}} \left[ \prod_{j=1}^d h(x_j|s_i) \right] P(s_i) \right). \quad (2.6) \end{aligned}$$

But this means that the summation in Equation 2.1 can be performed efficiently. We note that Condition 2.4 is sufficient to ensure the efficiency of other probabilistic inferences, such as the posterior of a hidden factor  $s_i$ :

$$P(s_i|\mathbf{x}) \sim P(s_i) \prod_{j=1}^d h(x_j|s_i). \quad (2.7)$$

### 2.3 Factorization via variational approximation

The Equation 2.3 for  $P(x_j|\mathbf{s})$  does not factorize for  $x_j = 1$ . Thus, in general, it is impossible to compute  $P(\mathbf{x})$  efficiently. To address this problem we approximate  $P(x_j|\mathbf{s})$  for  $x_j = 1$  with a factored variational lower bound used by Jaakkola and Jordan [11] in fully observable settings:

$$\begin{aligned} P(x_j = 1|\mathbf{s}) & \quad (2.8) \\ & \geq \prod_{i=1}^K \exp \left\{ q_j(i) s_i \left[ \log(1 - e^{-\theta_{0j} - \frac{\theta_{ij}}{q_j(i)}}) \right. \right. \\ & \quad \left. \left. - \log(1 - e^{-\theta_{0j}}) \right] + q_j(i) \log(1 - e^{-\theta_{0j}}) \right\}, \end{aligned}$$

where  $q_j$ s represent sets of variational parameters defining a multinomial distribution. Each component  $q_j(i)$  of the distribution can be viewed as a responsibility of a latent factor  $s_i$  for observing  $x_j = 1$ .

Incorporating the variational bound in the first part of Equation 2.3 we can obtain approximations  $\tilde{P}(\mathbf{x}|\mathbf{s}, \Theta, \mathbf{q}) \leq P(\mathbf{x}|\mathbf{s}, \Theta)$ ,  $\tilde{P}(\mathbf{x}, \mathbf{s}|\Theta, \mathbf{q}) \leq P(\mathbf{x}, \mathbf{s}|\Theta)$  and  $\tilde{P}(\mathbf{x}|\Theta, \mathbf{q}) \leq P(\mathbf{x}|\Theta)$  that factorize along latent factors  $s_i$ .

### 3 Learning of Noisy-or Networks with Hidden Units

The problem of learning of noisy-or bipartite networks has been addressed only in fully observable settings, that is, when both sources and observations are known. The learning methods take advantage of the decomposition of the model created by the introduction of special hidden variables. EM algorithm is then used to estimate the parameters of the modified network, which translate directly into the parameters of the original model. A reader interested in these transformations may consult papers by Heckerman [6], Vomlel [19] or Diez and Galan [8].

**3.1 Standard EM learning** Learning of the latent factor version of the Noisy-or network is much harder. Let  $D = \{\mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^N\}$  be a set of independent identically distributed samples of observable variables. Our objective is to find parameters  $\Theta$  that maximize the likelihood of the data,  $P(D|\Theta)$ . A standard approach to learn the parameters of the model in the presence of hidden variables is to use the Expectation-Maximization (EM) algorithm [7]. The EM computes the parameters iteratively by taking the following parameter update step:

$$\Theta^* = \arg \max_{\Theta} \sum_{n=1}^N \langle \log P(\mathbf{x}^n, \mathbf{s}^n|\Theta) \rangle_{P(\mathbf{s}^n|\mathbf{x}^n, \Theta')}$$

where  $\Theta'$  denotes previous-step parameters.

The main problem in applying the EM to our noisy-or model is that the joint distribution over every ‘‘completed’’ sample  $P(\mathbf{x}^n, \mathbf{s}^n|\Theta)$  does not decompose along hidden factors  $s_i$  and its expectation  $\langle \log P(\mathbf{x}^n, \mathbf{s}^n|\Theta) \rangle_{P(\mathbf{s}^n|\mathbf{x}^n, \Theta')}$  requires to iterate over all possible factor configurations. This is unfeasible since the configuration space grows exponentially in the number of factors. To alleviate this problem we optimize the parameters using the variational learning framework.

**3.2 EM for variational learning** The idea is to approximate the likelihood terms with their imprecise, but structurally more convenient surrogates. Additional set of free variational parameters is introduced to offer more flexibility and tune the approximation to specific settings. In our model, we substitute true conditional probabilities  $P(\mathbf{x}^n|\mathbf{s}^n, \Theta)$  that do not factorize, with

their factored lower-bound variational approximation  $\tilde{P}(\mathbf{x}^n | \mathbf{s}^n, \Theta, \mathbf{q}^n)$  as described in Section 2.3. Note that every datapoint  $\mathbf{x}^n$  comes with a different set of variational parameters  $\mathbf{q}^n$ .

In maximum likelihood learning we optimize the loglikelihood  $\log P(D|\Theta)$ . In our variational approach we optimize its lower bound:

$$\log \tilde{P}(D|\Theta, \mathbf{q}) = \sum_{n=1}^N \log \tilde{P}(\mathbf{x}^n | \Theta, \mathbf{q}^n)$$

The new quantity  $\log \tilde{P}(D|\Theta, \mathbf{q})$  depends on both parameters of the noisy-or model  $\Theta$  as well as on the variational parameters  $\mathbf{q}$ . Although we are ultimately interested in optimizing  $\Theta$  and variational parameters only play an auxiliary role, from the viewpoint of optimization of  $\log \tilde{P}(D|\Theta, \mathbf{q})$  there is no difference between the two and they must be treated the same way. Such an optimization can be carried out within the EM framework. In particular, the quantity can be maximized by iteratively reoptimizing  $(\Theta, \mathbf{q})$  pairs:

$$(\Theta, \mathbf{q})^* = \arg \max_{\Theta, \mathbf{q}} \sum_{n=1}^N \langle \log \tilde{P}(\mathbf{x}^n, \mathbf{s}^n | \Theta, \mathbf{q}^n) \rangle, \quad (3.9)$$

where  $\langle \cdot \rangle$  denotes the expectation, in this case taken over  $P(\mathbf{s}^n | \mathbf{x}^n, \Theta', \mathbf{q}'^n)$  and

$$\tilde{P}(\mathbf{x}^n, \mathbf{s}^n | \Theta, \mathbf{q}^n) = \tilde{P}(\mathbf{x}^n | \mathbf{s}^n, \Theta, \mathbf{q}^n) P(\mathbf{s}^n | \Theta) \quad (3.10)$$

$$P(\mathbf{s}^n | \mathbf{x}^n, \Theta', \mathbf{q}'^n) = Q'(\mathbf{s}^n) = \frac{\tilde{P}(\mathbf{x}^n, \mathbf{s}^n | \Theta, \mathbf{q}^n)}{\tilde{P}(\mathbf{s}^n | \Theta', \mathbf{q}'^n)},$$

and  $\Theta', \mathbf{q}'^n$  represent values of the parameters in the previous step. To simplify the notation in the rest of the paper, we use  $Q'(\mathbf{s}^n)$  to denote the posterior on hidden factors given the previous-step parameter values. Note that even if the  $\tilde{P}$  quantities are not probabilities, the posterior  $Q'(\mathbf{s}^n)$  is.

**3.3 Factorization of Expectations** Thanks to the factored form of  $\tilde{P}(\mathbf{x}^n | \mathbf{s}^n, \Theta, \mathbf{q}^n)$ , optimization steps in Equation 3.9 do not require us to iterate explicitly over all possible hidden factor configurations. More specifically, by substituting the expressions for  $\tilde{P}(\mathbf{x}^n, \mathbf{s}^n | \Theta, \mathbf{q}^n)$  and by taking the expectation in terms of the posterior  $Q'(\mathbf{s}^n)$  we obtain:

$$\begin{aligned} & \langle \log \tilde{P}(\mathbf{x}^n, \mathbf{s}^n | \mathbf{q}) \rangle_{Q'(\mathbf{s}^n)} \\ &= \left[ \sum_{i=1}^K \langle s_i^n \rangle_{Q'(\mathbf{s}^n)} \log \frac{\pi_i}{(1 - \pi_i)} + \log(1 - \pi_i) \right] + \\ &+ \left[ \sum_{j=1}^d \left( \sum_{i=1}^K -\langle s_i^n \rangle_{Q'(\mathbf{s}^n)} - \theta_{ij}(1 - x_j^n) \right) - \theta_{0j}(1 - x_j^n) \right] \end{aligned} \quad (3.11)$$

$$\begin{aligned} &+ \sum_{j=1}^d \sum_{i=1}^K \left[ \langle s_i^n \rangle_{Q'(\mathbf{s}^n)} q_j^n(i) x_j^n \log \left( 1 - e^{-\theta_{0j} - \frac{\theta_{ij}}{q_j^n(i)}} \right) \right. \\ &+ \left. (1 - \langle s_i^n \rangle_{Q'(\mathbf{s}^n)}) q_j^n(i) x_j^n \log(1 - e^{-\theta_{0j}}) \right] \end{aligned}$$

We see that for our factored approximation, the expectations are easy and the computations boil down to taking expectations over individual factors. Since the hidden factors take on binary values 0 and 1, the expectations for individual factors are just their probabilities of assuming value 1 and can be calculated using Equation 2.7.

**3.4 Parameter optimizations in EM** In every cycle of the EM algorithm we must reoptimize both the parameters  $\Theta$  and all variational parameters  $\mathbf{q}^n$ , one set per every data point. Unfortunately, no closed form solution for this task exists. We resort to iterative solutions, where parameters  $\mathbf{q}^n$  and  $\Theta$  are updated (optimized) until convergence.

We apply numerical and iterative optimization techniques to obtain partial solutions. However, we note that the dependencies among parameters to be optimized are relatively sparse and optimizations can be often performed quite efficiently. In particular, the iterative formula for a variational parameter  $q_j^n(i)$  only involves  $q_j^n(i)$  itself. We are dealing with  $DK$  instances of one-dimensional optimization for each datapoint, rather than with optimization in a higher-dimensional space.

Complete parameter update formulas we derived and use in our procedure are summarized in Figure 2. The updates were derived by calculating partial derivatives of the objective function and setting them to 0.

The precise analysis of algorithm's time complexity would be a tedious undertaking as it involves considerations of the convergence rates of nested iterative procedures. We demonstrate experimentally that the approximation yields a tractable algorithm.

**3.5 Regularization effect** While testing our variational learning algorithm we noticed its ability to automatically shut off "unused" noisy-or links. This suggests the presence of an inherent regularization correction. Examining the objective function (Equation 3.9) and optimization updates (in Figure 2) we can see that there is indeed a "natural" tendency of the method to drive unused parameters to 0, due to the presence of the term:  $-\langle s_i^n \rangle_{Q'(\mathbf{s}^n)} \theta_{ij}(1 - x_j^n)$  in the objective function in Equation 3.9. The term can be viewed as a regularization penalty assigned to large values of  $\theta_{ij}$  if these are not supported by data. Intuitively, the link with a poor

**Updates of variational parameters  $q_j^n(i)$  (one per sample).** Iterate until fixpoint:

$$q_j^n(i) \leftarrow \langle s_i^n \rangle_{Q'(s)} \frac{1}{\log(1 - e^{-\theta_{0j}})} \left[ q_j^n(i) \log(1 - A^n(i, j)) - \theta_{ij} \frac{A^n(i, j)}{1 - A^n(i, j)} - q_j^n(i) \log(1 - e^{-\theta_{0j}}) \right] \quad (3.12)$$

subject to condition  $\sum_{i=1}^K q_j^n(i) = 1$  assured through the normalization step.  $A^n(i, j)$  stands for  $e^{-\theta_{0j} - \frac{\theta_{ij}}{q_j^n(i)}}$ .

**Updates of  $\theta_{ij}$ s.** Search for the root of  $\partial\mathcal{F}/\partial\theta_{ij}$  via a numerical method:

$$\sum_{n=1}^N \langle s_i^n \rangle_{Q'(s)} \left[ -1 + x_j^n \frac{1}{1 - A^n(i, j)} \right] = 0 \quad (3.13)$$

**Updates  $\theta_{0j}$ s.** Search for the root of  $\partial\mathcal{F}/\partial\theta_{0j}$  via a numerical method:

$$\sum_{n=1}^N \left[ \sum_{i=1}^K \langle s_i^n \rangle_{Q'(s)} q_j^n(i) x_j^n \left( \frac{A^n(i, j)}{1 - A^n(i, j)} - \frac{e^{-\theta_{0j}}}{1 - e^{-\theta_{0j}}} \right) \right] + \left[ -(1 - x_j^n) + \sum_{i=1}^K x_j^n q_j^n(i) \frac{e^{-\theta_{0j}}}{1 - e^{-\theta_{0j}}} \right] = 0 \quad (3.14)$$

**Updates of  $\pi_i$ s:**

$$\pi_i = \frac{1}{N} \sum_{n=1}^N \langle s_i^n \rangle_{Q'(s)} \quad (3.15)$$

Figure 2: A summary of iterative optimization steps for the variational learning method

support in the data is shut down to avoid the penalty.

#### 4 Evaluation of the variational learning algorithm

To analyze the performance of our variational algorithm, we applied it first to an image deconvolution problem. In this problem, we use a bipartite noisy-or network with 8 hidden sources. Each source is associated with an  $8 \times 8$  image pattern. The patterns are shown in Figure 3. If the source is active (set to 1) its noise-corrupted pattern is projected to the output. The patterns from multiple sources (if they are active) and the leak pattern are combined using noisy-or units to generate the output image. The image patterns and their associated noise components are defined fully by the parameters of the noisy-or model.

We used the above noisy-or model to generate a set of training images. Figure 4 shows examples of 16 convoluted images generated by the model. The learning objective was to estimate and recover the distribution of the original model purely from the observational data – the noise-corrupted convoluted images. In order to assess the characteristics of our variational algorithm we run two sets of experiments, observing the quality of the solution and its running time while varying (1) the number of samples and (2) the number of assumed latent sources.

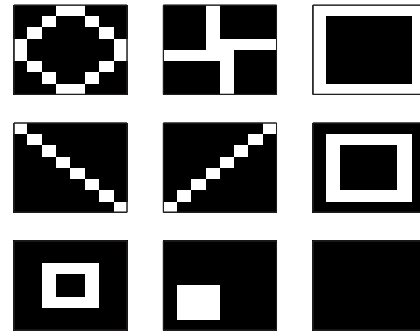


Figure 3: Image patterns associated with hidden sources used in the image deconvolution problem. The ninth (bottom-right) pattern corresponds to the leak.

**4.1 Effect of the sample size** We used the noisy-or network with 8 hidden sources and image patterns from Figure 3 to generate datasets with 50 - 5000 examples. These samples were then given to the learning algorithm. The learning process always starts from the complete network, no structure relating the sources and observables is given. The new (learned) model was evaluated in terms of: (1) Comparison of learned source images to original images (2) Data

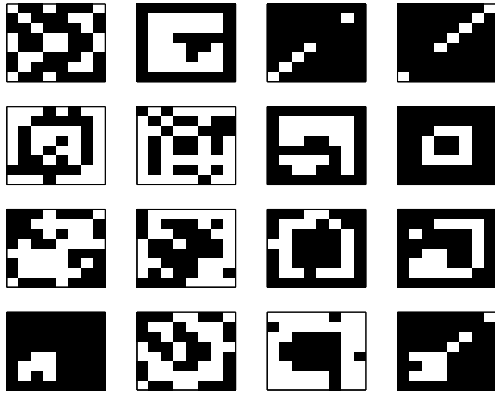


Figure 4: Example images generated by the latent noisy-or model with parameters corresponding to patterns in Figure 3.

reconstruction error.<sup>1</sup>

Figure 5 shows the parameters of three noisy-or models recovered by the learning algorithm for varied sample sizes. It is apparent from the figure that larger number of samples lead to progressively improving models that are closer to the original model and approximates its patterns better. The model learned from 50 samples is cumbered with high variance brought about by the low number of training examples, but nevertheless it begins to capture some of the original source patterns. Sample sizes of 500 and 1000 improve the pattern recovery. For 1000 samples we were able to recover almost all sources used to generate data with relatively small distortion. Naturally, inherent stochasticity will cause the sources to differ slightly in each run of the algorithm.

Latent variable models are very useful in dimensionality reduction. Given the learned noisy-or model and an image observed on the output, one can compute the posterior of each hidden source and pick the value (0 or 1) that comes with the higher posterior probability. Hidden sources and their 0/1 values then act as a low-dimensional representation of the data. High-dimensional data can be recovered back by sampling the output according to hidden source values and the pa-

<sup>1</sup>Note that it is very difficult to apply standard distance measures for distributions, such as KL-divergence or Hellinger's distance, to evaluate and compare two high-dimensional multivariate distributions. In our case, it would require to compute and compare probabilities of  $2^{64}$  possible image configurations. Approximate distance measures based on corresponding empirical distributions obtained via sampling suffer from a similar problem: it is extremely difficult to achieve an overlap carrying a significant probability mass between the supports of the respective empirical distributions.

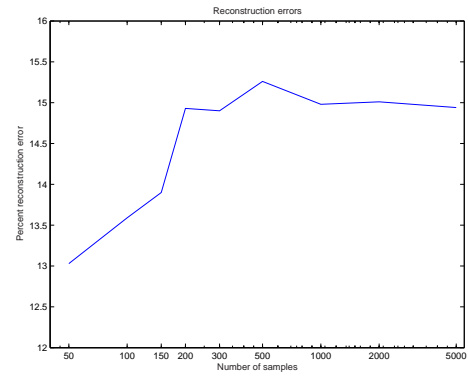


Figure 6: Reconstruction errors obtained from the learning algorithm for varied sample-sizes, averaged over 50 trials.

rameters of the noisy-or model. The difference between the original data point and its reconstruction after the initial reduction defines the reconstruction error. In our case, the reconstruction error is computed as portion of bits in which the original data differs from the reconstructed data.<sup>2</sup>

Figure 6 illustrates the reconstruction error of the model learned by the variational algorithm for different sample sizes. We clearly see the reconstruction error is smaller for very small sample sizes and stabilizes for sample sizes over 200. This can be explained by overfitting of the model for small sample sizes, and the saturation of the model to its stochastic limit for larger sample sizes.

The running time of the variational algorithm for different sample sizes is shown in Figure 7. The nearly straight line plotted indicates that the complexity of the algorithm grows polynomially with the number of samples. Indeed, we have observed that the time complexity scales approximately linearly with the number of samples. There appears to be no statistically significant effect of sample size on the number of EM iterations the algorithm performs.

**4.2 Model selection** In real-world data, the correct number of hidden sources is only rarely known in advance. Then the important question is whether the correct number of sources can be determined automatically by the learning algorithm. To analyze this aspect of the problem we run a series of learning experiments on models with different number of latent sources. To assure

<sup>2</sup>To assess the significance of the learning error, consider that the training sets used contain on average approximately 32% of 1s. Therefore, the trivial majority-class reconstruction baseline would achieve that error.

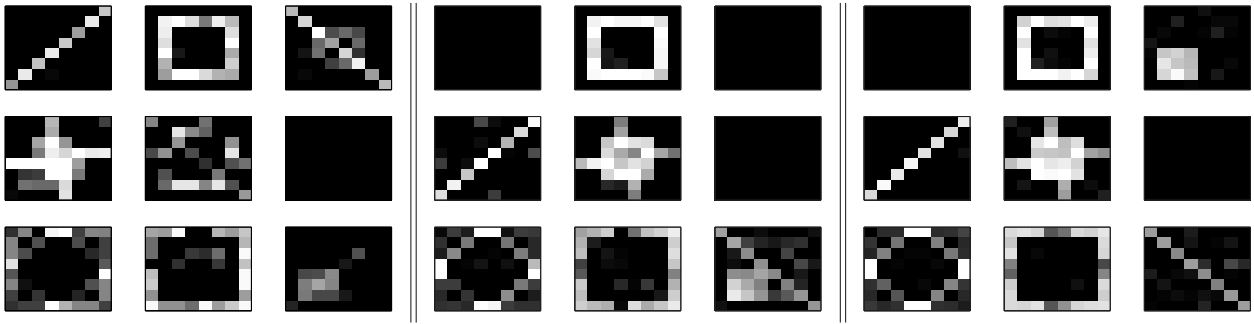


Figure 5: Examples of models learned from 50, 200 and 1000 samples (from left to right). The differences among models illustrate the improvement in the model recovery for increasing sample size. Although some source images are visibly identified with as few as 50 samples, the noise in many images is apparent. Models learned from 200 and 1000 samples are improved. Contrasting 200-sample model to 1000-sample model, a source image stepped out of the leak factor (top row, right column). Additionally, the sources have stabilized, “shadows” were cleaned (compare the source in left column, second row). The only flaw to the 1000-sample model is the source in the center which captured two of the original sources.

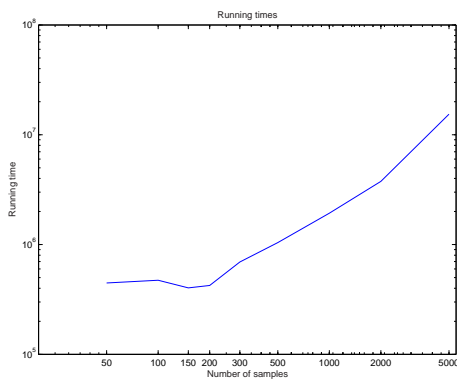


Figure 7: Runtimes of the algorithm, averaged over 50 trials. Considering the wide range of sample sizes tested, we plot the runtimes on a log-log plot.

a fair comparison, the dataset used to train the models was fixed over the course of the experiment.

The results are summarized in Figure 8. The reconstruction error plot demonstrates that as we increase the number of latent sources the learner takes advantage of all sources available to it at 6 sources or fewer, then starts to plateau at 8 sources. This agrees well with the number of latent sources used to generate the data.

To assess the recovery performance, we looked at patterns learned by the algorithm, much like those in Figure 5 and counted the number of identified sources. The inspection of the learned models showed that the number recovered sources levels out at around 7, other sources were shut down via regularization

effects (Section 3.5). Taking into account the existence of the leak node (which effectively adds one source), this matches or is very close to the true number of sources. Taking advantage of these phenomena one does not have to identify the number of hidden sources in advance, the algorithm finds a reasonable estimate of the correct number on its own at only minor additional computational cost.

The analysis of running times for different number of sources in Figure 8 shows that the runtimes scale roughly linearly with the number of assumed latent sources. This gives an empirical support for the efficiency of variational EM approximation as compared to the exponential complexity of the exact EM algorithm with respect to the number of sources.

## 5 Noisy-or component analysis of citation data

To show the benefit of our model in a real-world application we applied the model to perform *component analysis* of a citation dataset derived from online publications in the area of Machine learning. The dataset was built from approximately 17,000 hypertext documents from the CiteSeer service. We chose 40 prominent authors in the field of Statistical Machine Learning, to limit ourselves to a domain where we can confidently assess the soundness of the obtained results. The data were then processed into a binary matrix. This matrix contains 1 at position  $(i, j)$  if the document  $i$  cites author  $j$ .

The noisy-or model fits well the structure this dataset exhibits. A contemporary paper in this field is likely to touch upon several topics and combine or improve on them. We would expect the hidden factors to roughly match the paper keywords, each topic factor

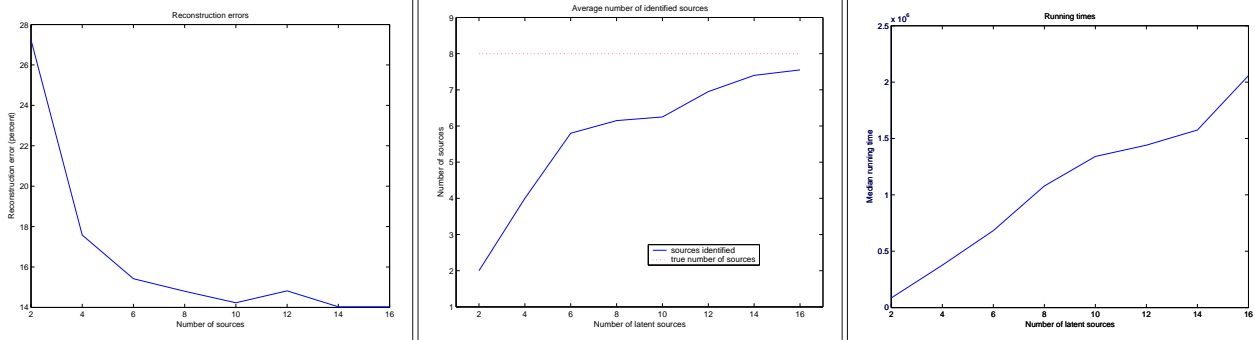


Figure 8: Average reconstruction error (left panel), average number of identified sources (middle) and median running times (right) plotted against the number of assumed latent sources. The red dotted line in the middle plot represents the true number of sources (8). The statistics in this figure were obtained from 25 experimental runs.

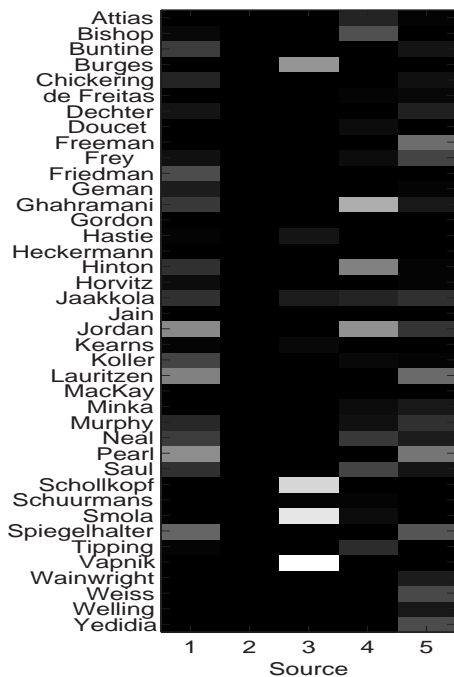


Figure 9: The result of the noisy-OR component analysis on the citation dataset. The columns visualize the parameters of the noisy-or loading matrix after they are rescaled by the prior of the source. Black fields correspond to 0s in the loading matrix, while white would correspond to 1s.

having its seminal papers whose authors thereby become likely to be cited.

We ran our noisy-or model on the CiteSeer dataset using 5 hidden sources. Figure 9 illustrates the outcome of the analysis. The obtained results indicate the presence of the following components:

- The authors dominating the first component are: J. Pearl, M. Jordan, S. Lauritzen and D. Spiegelhalter. Weaker ties are to W. Buntine, N. Friedman and D. Koller. This component discovered many respected authors of basic references and tutorials on Bayesian belief networks.
- The second source was shut down as the algorithm did not reveal any other interesting group in this run.
- C. Burges, B. Schölkopf, A. Smola and V. Vapnik form the core of the third component. Without any doubt, this component represents the kernel and SVM research community.
- The authors prominent in the fourth factor are Z. Ghahramani, M. Jordan, G. Hinton, R. Neal, L. Saul, C. Bishop and M. Tipping. This source captures the variational approximation community.
- The last component consists of the following authors: B. Frey, W. Freeman, K. Murphy, S. Lauritzen, J. Pearl, Y. Weiss and J. Yedidia. All authors published extensively on loopy belief propagation, using J. Pearl's BP algorithm. The presence of an outlier in this set, S. Lauritzen, can be attributed to the fact that he is among the most frequently cited authors in the general context of Bayesian networks. Conclusively, we can say our algorithm found the LBP community.

The results obtained for the citation data show the potential benefit of the noisy-or model and its ability to uncover semantically sound component structure in the binary data. We note there is a conceptual difference between the noisy-or model and mixture models, such as the aspect model [10], used frequently in the analysis of documents. The key difference is that the aspect model assigns each document a convex combination of topic factors, while our model computes a vector of binary indicators, each corresponding to one topic. Each model stresses a different type of the structure and both analyses can complement each other to improve the understanding of the data at hand.

## 6 Conclusions

We have devised and presented an EM-based variational algorithm for learning latent factor models with noisy-or units. The algorithm alleviates the key limitation of exact learning algorithms – their exponential dependency on the number hidden factors. The proposed variational algorithm makes no assumption about the structure of the underlying noisy-or network, the structure is fully recovered during the learning process.

We tested the algorithm on two problems: (1) image deconvolution problem and (2) analysis of citation data. The algorithm showed a good scale-up potential with a very good model recovery and error reconstruction performances on the image problem. On citation data it successfully discovered components that represent established communities. We demonstrated how the noisy-or latent variable model offers itself as a tool of inquiry of social networks and internet communities.

An in-depth comparison of the noisy-or component analyzer to alternative component analysis frameworks, most importantly Probabilistic Latent Semantic Analysis, remains an interesting open problem and a focus of our continued research interest.

## 7 Acknowledgements

This research was supported in part by the Research Development Fund Award 36851 from the University of Pittsburgh and by National Science Foundation grants CMS-0416754 and ITR-0325353.

## References

- [1] Hagai Attias. Independent Factor Analysis. *Neural Computation*, 11(4):803–851, 1999.
- [2] Christopher M. Bishop. Latent variable models. In Michael I. Jordan, editor, *Learning in Graphical Models*, pages 371–403. MIT Press, 1999.
- [3] Christopher M. Bishop. Variational principal components. In *Proceedings of Ninth International Conference on Artificial Neural Networks*, volume 1, pages 509–514. ICANN, 1999.
- [4] D. Blei, A. Ng, and M. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, JAN 2003.
- [5] W. Buntine. Variational extensions to EM and multinomial PCA. In *ECML 2002*, 2002.
- [6] Heckerman David. Causal independence for knowledge acquisition and inference. In *Proc. of 9th Conf. on UAI93*, San Francisco, CA, 1993. Morgan Kaufmann Publishers.
- [7] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood for incomplete data via the EM algorithm. *Journal of Royal Statistical Society*, 39:1–38, 1977.
- [8] Francisco Diez and Severino Gallan. Efficient computation for the noisy max. *International Journal of Intelligent Systems*, 2003.
- [9] Zoubin Ghahramani and Michael I. Jordan. Factorial hidden Markov models. In David S. Touretzky, Michael C. Mozer, and Michael E. Hasselmo, editors, *Proceedings of Advances in Neural Information Processing Systems, NIPS*, volume 8, pages 472–478. MIT Press, 1995.
- [10] Thomas Hofmann. Probabilistic latent semantic analysis. In *Proc. of Uncertainty in Artificial Intelligence, UAI'99*, Stockholm, 1999.
- [11] Tommi Jaakkola and Michael I. Jordan. Variational probabilistic inference and the QMR-DT network. *Journal of Artificial Intelligence Research*, 10:291–322, 1999.
- [12] Michael Kearns and Yishay Mansour. Exact inference of hidden structure from sample data in noisy-OR networks. In *Proc. 14th Conf. on UAI98*, pages 304–310, 1998.
- [13] Xinghua Lu, Milos Hauskrecht, and Roger S. Day. Modeling cellular processes with variational bayesian cooperative vector quantizer. In *Pacific Symposium on Biocomputing (PSB)*, page to appear, 2004.
- [14] David MacKay. Probable networks and plausible predictions - a review of practical Bayesian methods for supervised neural networks. *Network: Computation in Neural Systems*, 6(3):469–505, 1995.
- [15] James W. Miskin. *Ensemble Learning for Independent Component Analysis*. PhD thesis, Selwyn College, University of Cambridge, 2000.
- [16] D. Ross and R. Zemel. Multiple cause vector quantization. In *Advances in Neural Information Processing Systems*, 2002.
- [17] M. Shwe, B. Middleton, D. Heckerman, M. Henrion, E. Horvitz, H. Lehmann, and G. Cooper. Probabilistic diagnosis using a reformulation of the INTERNIST-1/QMR knowledge base I. The probabilistic model and inference algorithms. *Methods of Information in Medicine*, 30:241–255, 1991.
- [18] Michael Tipping and Christopher Bishop. Probabilistic principal component analysis. Technical Report NCRG/97/010, Neural Computing Research Group,

Aston University, September 1997.

- [19] Jiří Vomlel. Noisy-or classifier. In *Proceedings of the 6th Workshop on Uncertainty Processing (WUPES 2003)*, pages 291–302, September 2003.