Journal of Machine Learning Research 7 (2006) 2189–2213 Submitted 01/2006; Published 10/2006

Noisy-OR Component Analysis and its Application to Link Analysis

Tomáš Šingliar TOMAS@CS.PITT.EDU
Miloš Hauskrecht MILOS@CS.PITT.EDU
Department of Computer Science
5329 Sennott Square
University of Pittsburgh
phone: (412) 624-8845

Editor: David M. Chickering

Abstract

We develop a new component analysis framework, the *Noisy-Or Component Analyzer* (NOCA), that targets high-dimensional binary data. NOCA is a probabilistic latent variable model that assumes the expression of observed high-dimensional binary data is driven by a small number of hidden binary sources combined via noisy-or units. The component analysis procedure is equivalent to learning of NOCA parameters. Since the classical EM formulation of the NOCA learning problem is intractable, we develop its variational approximation. We test the NOCA framework on two problems: (1) a synthetic image-decomposition problem and (2) a co-citation data analysis problem for thousands of CiteSeer documents. We demonstrate good performance of the new model on both problems. In addition, we contrast the model to two mixture-based latent-factor models: the probabilistic latent semantic analysis (PLSA) and latent Dirichlet allocation (LDA). Differing assumptions underlying these models cause them to discover different types of structure in co-citation data, thus illustrating the benefit of NOCA in building our understanding of high-dimensional datasets.

Keywords: Component analysis, Vector quantization, Variational learning, Link analysis

1. Introduction

Latent variable (or *latent factor*) models (MacKay, 1995; Bishop, 1999a) provide an elegant framework for modeling dependencies in high-dimensional data. Suppose that two observed random variables x_i, x_j are marginally dependent. A latent variable model explains their dependency by positing the presence of a hidden variable s representing their common cause. Examples of latent factor models include probabilistic principal component analysis (Tipping and Bishop, 1997; Bishop, 1999b), mixtures of factor analyzers (Attias, 1999), multinomial PCA (or *aspect*) models (Buntine, 2002; Hofmann, 1999a; Blei et al., 2003), the multiple cause model (Ghahramani and Jordan, 1995; Ross and Zemel, 2002) and independent component analysis frameworks (Attias, 1999; Miskin, 2000). The models are most often used for component analysis, where we want to identify a small number of underlying components (factors, sources, or signals) whose effects combine to form the observed data. Once a model is learned, it can be used to make inferences on hidden factors, such as to identify the document topics in the aspect model (Hofmann, 1999a; Blei et al., 2003) or regulatory signals in the microarray DNA data (Lu et al., 2004). In addition to their role in understanding the

structure of high-dimensional data, latent factor models can be applied in dimensionality reduction, where the hidden factor values are a low-dimensional representation of the data sample.

Factor and principal component analysis methods (Bartholomew and Knott, 1999; Jolliffe, 1986) and other component analysis frameworks (Attias, 1999) are traditionally applied to high-dimensional continuous-valued data. More recently, multinomial mixture models (Hofmann, 1999a; Blei et al., 2003) were shown to handle many-valued discrete variables successfully. However, component analysis methods specifically tailored to binary data remain scarce. In this work, we investigate a latent factor model designed for analysis of high-dimensional *binary* data. The dependencies between observables are represented using a small number of hidden binary factors whose effects are combined through noisy-or units. We therefore refer to the model as to "noisy-or component analyzer" (NOCA). Binary variables can, for instance, represent failures or congestions in transportation networks, spread of disease in epidemiology, or the presence of a link in a citation graph.

The principal limitation of latent factor models is the complexity of their learning (or *parameter estimation*), as the standard EM formulation becomes exponential in the number of hidden factors. To address the problem, we adopt a variational inference algorithm for bipartite noisy-or (B2NO) networks (Jaakkola and Jordan, 1999) and derive the corresponding learning algorithm for the model with hidden sources.

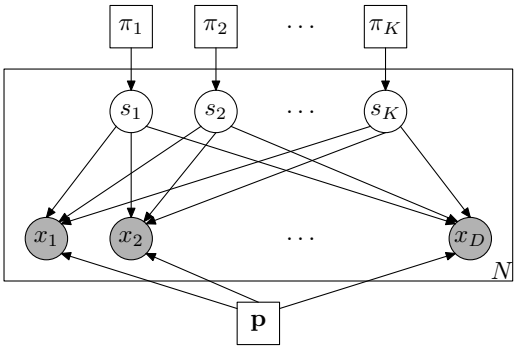
Two aspects of the new method are evaluated: (1) the quality of the approximate learning algorithm and (2) the adequacy of the model for real-world data. We use two different datasets to evaluate NOCA and its learning algorithm: a synthetic image-decomposition problem and a co-citation data analysis problem. The knowledge of the underlying model and hidden factors in the first problem (image data) enables us to assess the performance of the learning algorithm and its ability to recover the model. We judge the quality of the recovery both qualitatively and quantitatively in terms of the likelihood of test data and data reconstruction error. Running-time analysis verifies the expected polynomial scale-up.

The second evaluation problem is an application of NOCA to link and citation analysis. Citation data from over 6000 CiteSeer documents were extracted and analyzed with NOCA. To measure how well NOCA's hidden sources capture the cocitation relationships, we use a cosine-distance based metric and an inspection by a human judge. Perplexity of the testing set is used to gauge the predictive power of the learned model. NOCA results are compared to mixture-based latent variable models, represented by probabilistic latent semantic analysis (Hofmann, 1999a; Cohn and Chang, 2000) and its Bayesian extension – latent Dirichlet allocation (Blei et al., 2003). The mixture models view a document differently from NOCA. In consequence, each model class sees different facets of the data structure. NOCA's benefit is in the discovery of publication subcommunities in the data that the mixture models tend to overlook.

2. Noisy-OR Component Analysis

Technically, the noisy-or component analysis (NOCA) is a latent variable model with binary variables defined by a bipartite belief network structure in Figure 1.

The nodes in the top layer represent a vector of latent factors $\mathbf{s} = \{s_1, s_2, \ldots, s_K\}$ ("sources") with binary values $\{0,1\}$ and the nodes in the bottom layer an observable vector of binary features $\mathbf{x} = \{x_1, x_2, \ldots, x_D\}$. The connections between the two layers represent dependencies among the observables: the nodes coupled by a latent factor can exhibit a local dependency pattern. Parameter-



Notation:
D – observable dimensionality
K – latent dimensionality, $D > K$
N – number of datapoints
$\mathbf{x}$ – observables, indexed by j: x_j
$\mathbf{s}$ – latent sources, indexed by i: s_i
Parameters (square nodes):
$\mathbf{p}$ – loading matrix (with leak terms)
$\{\pi_i\}$ – source priors

Figure 1: The NOCA model in plate notation. Shaded nodes correspond to observables. (In the entire text, boldface letters will denote vectors or matrices.)

izing the bottom-layer nodes with noisy-or units reduces the model's parameter space to $KD+K+D$ free parameters:

- a set of K prior probabilities π_i parameterizing the (Bernoulli) prior distributions $P(s_i)$ for every hidden factor s_i;
- a set of DK parameters $\mathbf{p} = \{p_{ij}\}_{j=1,\ldots,D}^{i=1,\ldots,K}$ of the noisy-or conditional probability tables, one for each pair of hidden factor i and observed component j.
- a set of D parameters p_{0j} representing "other causes." These can be incorporated into $\mathbf{p}$ by positing a latent factor s_0 with $p(s_0 = 1) = 1$, where notationally convenient.

The NOCA model resembles the QMR-DT model (Shwe et al., 1991) in the structure and type of nodes used. However, it is from the outset assumed to be fully connected. The model is simplified during learning by setting the weight of most connections to zero.[1] NOCA makes no assumption as to the interpretation of random variables. For example, although features might correspond to words when analyzing text documents; citation indicator variables will be used when analyzing references among scholarly articles.

2.1 The Joint Distribution over Observables

The joint probability of an observation vector $P(\mathbf{x})$ exemplifies and subsumes the probabilistic queries we need to evaluate. Given the bipartite model, $P(\mathbf{x})$ is obtained as

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

1. This is in contrast with the structure-learning algorithm proposed by Kearns and Mansour (Kearns and Mansour, 1998). Their algorithm is exponential in the maximum number of hidden factors contributing to any observable variable. Therefore, they limit the in-degree of the bottom layer nodes to obtain a polynomial algorithm. Our algorithm does not make any such structural assumption.

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 the noisy-or model:

$$P(x_j|\mathbf{s}) = \left[1-(1-p_{0j})\prod_{i=1}^{K}(1-p_{ij})^{s_i}\right]^{x_j}\left[(1-p_{0j})\prod_{i=1}^{K}(1-p_{ij})^{s_i}\right]^{(1-x_j)}, \quad (2)$$

where p_{0j} is the leak probability that models "all other" causes.

Equation 2 can be reparameterized with $\theta_{ij} = -\log(1-p_{ij})$ to obtain:

$$P(x_j|\mathbf{s}) = \exp\left[x_j \log\left(1-\exp\left\{-\theta_{0j}-\sum_{i=1}^{k}\theta_{ij}s_i\right\}\right)+(1-x_j)\left(-\theta_{0j}-\sum_{i=1}^{K}\theta_{ij}s_i\right)\right]. \quad (3)$$

This reparameterization will prove useful in the following description of the variational lower bound.

2.2 The Factorized Variational Bound

The bottleneck in computing the joint probability over observables, $P(\mathbf{x})$ in Equation 1, is the sum that ranges over all possible latent factor configurations. However, it is easy to see that if $P(x_j|\mathbf{s})$ for both $x_j = 0$ and $x_j = 1$ could be expressed in a factored form as:

$$P(x_j|\mathbf{s}) = \prod_{i=1}^{K} h(x_j|s_i), \text{ such that } \forall i,j : h(x_j|s_i) \geq 0, \quad (4)$$

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

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

Such decomposition would imply that the summation in Equation 1 can be performed efficiently. Note that the condition of Equation 4 is sufficient to ensure tractability of other inference queries, such as the posterior of a hidden factor s_i:

$$P(s_i|\mathbf{x}) \propto P(s_i)\prod_{j=1}^{d}h(x_j|s_i). \quad (5)$$

However, while Equation 3 defining $P(x_j|\mathbf{s})$ decomposes for $x_j = 0$, it does not factorize for $x_j = 1$. Thus, in general, it is impossible to compute $P(\mathbf{x})$ efficiently. We approximate $P(x_j|\mathbf{s})$ for $x_j = 1$ with a factored variational lower bound (Jaakkola and Jordan, 1999):

$$P(x_j=1|\mathbf{s}) \geq \quad (6)$$
$$\tilde{P}(x_j|\mathbf{s}) = \prod_{i=1}^{K}\exp\left\{q_j(i)s_i\left[\log(1-e^{-\theta_{0j}-\frac{\theta_{ij}}{q_j(i)}})-\log(1-e^{-\theta_{0j}})\right]+q_j(i)\log(1-e^{-\theta_{0j}})\right\},$$

where q_js 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 $x_j = 1$. If we denote the complex expression inside the product on the right-hand side of Equation 6 by $h(x_j|s_i)$, we have the sought-after decomposition.

Incorporating the variational bound into the first, nondecomposing term in Equation 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. The Variational Learning Algorithm

The key step of component analysis corresponds to the learning of the latent factor model from data. The problem of learning of bipartite noisy-or networks has been addressed only in the fully observable setting; that is, when both the sources and observations are known. The learning methods take advantage of the decomposition of the model created by the introduction of special hidden variables (Heckerman, 1993; Vomlel, 2003; Diez and Gallan, 2003). The EM algorithm is then used to estimate the parameters of the modified network, which translate directly into the parameters of the original model. However, to our knowledge, no learning algorithm for B2NO networks has been derived for the case of unobservable source layer.

In this section, we motivate and detail the derivation of the variational learning algorithm, following the EM-framework. We identify the crucial hurdles in deriving an efficient algorithm and show how the variational approximation overcomes them.

3.1 Classical EM Formulation

Let $D = \{\mathbf{x}^1, \mathbf{x}^2, \cdots \mathbf{x}^N\}$ be a set of N i.i.d. vectors of observable variables. Our objective is to find parameters Θ that maximize the likelihood of the data, $P(D|\Theta)$. The standard approach to learn the parameters of the model in the presence of hidden variables is the Expectation-Maximization (EM) algorithm (Dempster et al., 1977). 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 Θ' denotes previous-step parameters.

The main problem in applying the EM to the 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 thus its expectation $\langle \log P(\mathbf{x}^n, \mathbf{s}^n|\Theta) \rangle_{P(\mathbf{s}^n|\mathbf{x}^n,\Theta')}$ requires iteration over all possible latent factor configurations. This is infeasible since the configuration space grows exponentially in the number of factors. Note that even if we could solve the inference query $P(\mathbf{s}^n|\mathbf{x}^n,\Theta')$ efficiently, we still cannot push the expectations inward over the nonlinearities – we also need to decompose the term inside the expectation.

3.2 Variational EM

The idea of variational methods is to approximate the likelihood terms with their imprecise, but structurally more convenient surrogates. In summary, an additional set of free variational parameters $\mathbf{q}$ (Section 2.2) is introduced that offers the flexibility to perform more efficient calculations of

the joint and posterior distributions within the EM algorithm. In particular, we replace the 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 2.2. As a consequence, the approximate posterior $\tilde{P}(\mathbf{s}^n|\mathbf{x}^n,\Theta,\mathbf{q}^n)$ also factorizes, which simplifies the expectation step of the algorithm. The new EM algorithm iteration becomes:

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

where Θ' and $\mathbf{q}^{n'}$ denote previous-step model and variational parameters.

In ML learning, we maximize $\log P(D|\Theta)$ with respect to Θ. In NOCA, we maximize a lower bound on $\log P(D|\Theta)$ instead, to ease the computational complexity brought by hidden variables. First, let us simplify the expectation distribution – the hidden source posterior:

$$\begin{aligned}
\log P(D|\Theta) &= \log \prod_{n=1}^{N} P(\mathbf{x}^n|\Theta) \\
&= \sum_{n=1}^{N} \log \left[\sum_{\{\mathbf{s}^n\}} P(\mathbf{x}^n,\mathbf{s}^n|\Theta) \right] \\
&= \sum_{n=1}^{N} \log \left[\sum_{\{\mathbf{s}^n\}} P(\mathbf{x}^n,\mathbf{s}^n|\Theta,\mathbf{q}^n) \frac{Q(\mathbf{s}^n)}{Q(\mathbf{s}^n)} \right] \\
&\geq \sum_{n=1}^{N} \left[\sum_{\{\mathbf{s}^n\}} \langle \log P(\mathbf{x}^n,\mathbf{s}^n|\Theta) \rangle_{Q(\mathbf{s}^n)} - \langle \log Q(\mathbf{s}^n) \rangle_{Q(\mathbf{s}^n)} \right]
\end{aligned}$$

This lower bound follows from Jensen's inequality for any arbitrary distribution over the hidden sources $Q(H) = \prod_{n=1}^{N} Q(\mathbf{s}^n)$ (Jordan et al., 1999; Saul et al., 1996; Ghahramani and Jordan, 1997). However, even with a decomposable Q, we cannot take expectations of $\log P(\mathbf{x}^n,\mathbf{s}^n|\Theta)$ easily, because the noisy-or distribution is not in the exponential family and the s_i's reside inside nonlinearities. We apply Equation 6 to obtain a further lower bound:

$$\begin{aligned}
\log P(D|\Theta) &\geq \sum_{n=1}^{N} \left[\sum_{\{\mathbf{s}^n\}} \langle \log P(\mathbf{x}^n,\mathbf{s}^n|\Theta) \rangle_{Q(\mathbf{s}^n)} - \langle \log Q(\mathbf{s}^n) \rangle_{Q(\mathbf{s}^n)} \right] \\
&\geq \sum_{n=1}^{N} \left[\sum_{\{\mathbf{s}^n\}} \left\langle \log \tilde{P}(\mathbf{x}^n,\mathbf{s}^n|\Theta,\mathbf{q}^n) \right\rangle_{Q(\mathbf{s}^n)} - \langle \log Q(\mathbf{s}^n) \rangle_{Q(\mathbf{s}^n)} \right] \\
&= \sum_{n=1}^{N} \left[\sum_{\{\mathbf{s}^n\}} \left\langle \log \tilde{P}(\mathbf{x}^n|\mathbf{s}^n,\Theta,\mathbf{q}^n) P(\mathbf{s}^n|\Theta) \right\rangle_{Q(\mathbf{s}^n)} - \langle \log Q(\mathbf{s}^n) \rangle_{Q(\mathbf{s}^n)} \right] \\
&= \sum_{n=1}^{N} \mathcal{F}_n(\mathbf{x}^n, Q(\mathbf{s}^n)) \\
&= \mathcal{F}(D, Q(H)),
\end{aligned}$$

where $\mathbf{q}^n$ are parameters of the lower bound approximation described in Section 2.2.

The variational EM that optimizes the bound also proceeds, like standard EM, in two steps. The E-step computes the expectation distribution $Q(\mathbf{s}^n)$. We could in principle choose any distribution Q, but it is desirable to choose one that makes the variational bound as tight as possible. The variational bound of $\log P(D|\Theta)$ is the tightest at $Q(\mathbf{s}^n) = P(\mathbf{s}^n|\mathbf{x}^n,\Theta)$. Since that ideal posterior is intractable, we define $Q(\mathbf{s}^n)$ to be the tractable posterior probability $\tilde{P}(\mathbf{s}^n|\mathbf{x}^n,\Theta',\mathbf{q}^n)$, where Θ' are fixed previous step parameters and $\mathbf{q}^n$ are tuned to obtain the best approximation to the true posterior. (Alternatively, we could separately and explicitly optimize Q to maximize $\mathcal{F}(D,Q(H))$.)

The new $\mathbf{q}^n$ s are obtained that maximize $\tilde{P}(\mathbf{x}^n|\mathbf{s}^n,\Theta',\mathbf{q}^n)$ by an iterative procedure described in Figure 2. These iterative updates essentially form an embedded EM loop and are derived in (Jaakkola et al., 1996). The subsequent computation of $\tilde{P}(\mathbf{s}^n|\mathbf{x}^n,\Theta',\mathbf{q}^n)$ decomposes along the hidden factors and can be performed in linear time according to Equation 5. Obtaining the posteriors on hidden sources concludes the E-step.

The M-step optimizes $\mathcal{F}(D,Q(H))$ with respect to Θ. Given the decomposable $Q(\mathbf{s}^n)$, $\mathcal{F}_n(\mathbf{x}^n,Q(\mathbf{s}^n))$ can be rewritten as:

$$\begin{aligned}
\mathcal{F}_n(\mathbf{x}^n,Q(\mathbf{s}^n)) &= \left\langle \log \tilde{P}(\mathbf{x}^n|\mathbf{s}^n,\mathbf{q}^n,\Theta)\right\rangle_{Q(\mathbf{s}^n)} - \left\langle Q(\mathbf{s}^n)\right\rangle_{Q(\mathbf{s}^n)} \\
&= \left[\sum_{i=1}^{K} \langle s_i^n\rangle_{Q(s_i^n)} \log\frac{\pi_i}{(1-\pi_i)} + \log(1-\pi_i)\right] \\
&+ \left[\sum_{j=1}^{d}\left(\sum_{i=1}^{K} -\langle s_i^n\rangle_{Q(s_i^n)}\theta_{ij}(1-x_j^n)\right) - \theta_{0j}(1-x_j^n)\right] \qquad (7)\\
&+ \sum_{j=1}^{d}\sum_{i=1}^{K}\left[\langle s_i^n\rangle_{Q(s_i^n)} q_j^n(i) x_j^n \log\left(1-e^{-\theta_{0j}-\frac{\theta_{ij}}{q_j^n(i)}}\right) + \left(1-\langle s_i^n\rangle_{Q(\mathbf{s}_i^n)}\right) q_j^n(i) x_j^n \log(1-e^{-\theta_{0j}})\right] \\
&- \left\langle Q(\mathbf{s}^n)\right\rangle_{Q(\mathbf{s}^n)}
\end{aligned}$$

The last term is the entropy of the variational distribution, it does not depend on Θ and can be ignored in further M-step derivations.

For the rest of the paper all expectations are over $Q(\mathbf{s}^n)$ – the variational posterior on hidden factors based on previous-step parameters. The simplified notation leaves the dependence on $\mathbf{x}$ and $\mathbf{q}$ implicit, but also expresses the intuition that by replacing the posterior by a variational distribution, we effectively "disconnected" the model.

Since $\log P(\mathbf{x}^n,\mathbf{s}^n|\Theta)$, the term inside expectation, is approximated using the same transformation of $P(\mathbf{x}|\mathbf{s})$ as the posterior distribution over the hidden sources, the $\mathbf{q}$ computed in the E-step can be reused in the M-step. The parameter updates for M-step can be derived straightforwardly by setting

$$\frac{\partial}{\partial\theta_{ij}}\mathcal{F}(D,Q(H)) = 0 \qquad \frac{\partial}{\partial\theta_{0j}}\mathcal{F}(D,Q(H)) = 0.$$

Unfortunately, no closed form solutions for these tasks exist. We update the parameters Θ simultaneously by setting them to the numerical solutions of the above equations and iterate the updates until convergence. The numerical solutions are obtained by bisection search (Figure 3). The parameters are set to random non-zero values in the first EM iteration. We note that the dependencies among parameters are relatively sparse and optimizations typically converge in very few optimization steps. The complete parameter update formulas we derived and use in our procedure are summarized in Figure 2.

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

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

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

Updates of θ_{ij}s. Find the root of $\partial \mathcal{F} / \partial \theta_{ij} = 0$ numerically:

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

Updates of θ_{0j}s. Find the root of $\partial \mathcal{F} / \partial \theta_{0j} = 0$ numerically:

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

Updates of π_is:

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

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

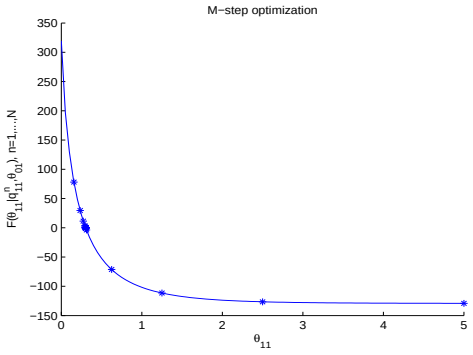
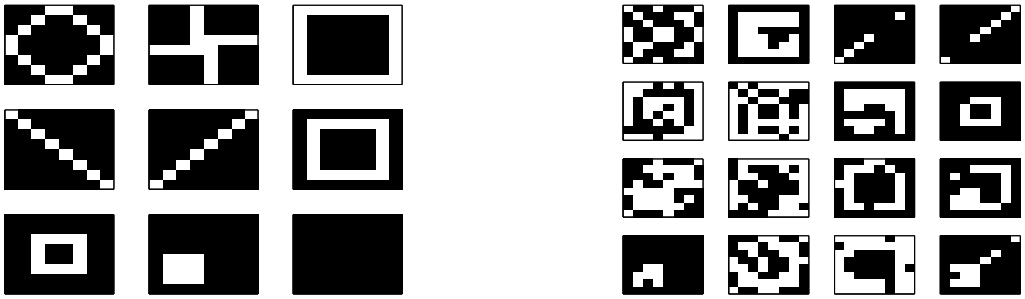


Figure 3: M-step optimization is simply a bisection-search procedure. The curve is the partial derivative of the objective function F w.r.t. θ_{11} plotted as a function of θ_{11}. The little stars on the curve represent iterations of the bisection algorithm. The advantages of the bisection algorithm come from its simplicity: no derivatives that would be costly to compute (we have to iterate through the data to compute F!) and good numerical stability. The search typically converges in few (~ 10) iterations.



(a) (b)

Figure 4: Model reconstruction experiments. **(a)** Image patterns associated with hidden sources used in the image decomposition problem. The ninth (bottom-right) pattern corresponds to the leak. **(b)** Example images generated by the NOCA model with parameters corresponding to patterns in panel (a).

3.3 Simplicity Bias

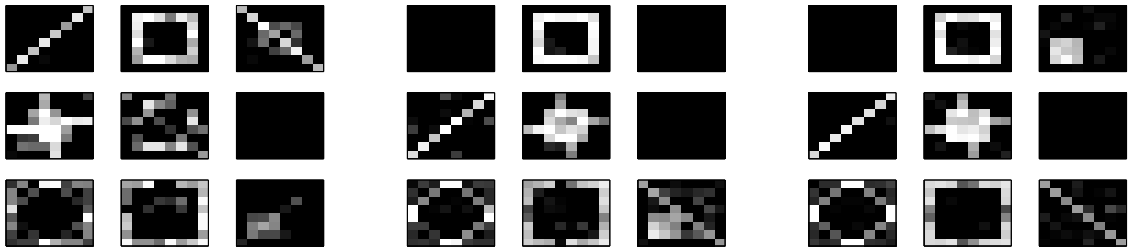
The empirical evaluation of the NOCA model revealed that the model is able to automatically shut off "unused" noisy-or links between sources and observations. This suggests the presence of a term encouraging sparse models in the functional $\mathcal{F}$. Indeed, the term: $-\langle s_i^n \rangle_{Q(\mathbf{s}^n)} \theta_{ij}(1-x_j^n)$ in Equation 7 can be viewed as a regularization-like penalty[2] assigned to large values of θ_{ij} if these are not supported by data. A penalty proportional to θ_{ij} and the posterior of a hidden source is added for each observable x_j^n that is equal to 0. This has an appealing intuitive interpretation: it is unlikely that the observation x_j is 0, if the source is on ($\langle s_i \rangle$ is high) and the link between s_i and x_j is strong $(\theta_{ij} >> 0)$. Consequently, the link in between the source j and observation i is driven to zero if not supported by the presence of positive observations. If all links between a source and observations are driven to zero, the source is effectively disconnected and can be pruned from the network. We demonstrate this effect in the experiments in Section 4.2.

4. Evaluation of NOCA

In this section, we will evaluate NOCA and its variational learning algorithm on a synthetic image dataset built using NOCA model. The advantage of using a synthetic dataset is that the true model as well as the instantiations of the hidden sources are known.

The image datasets used in the experiments are created by sampling from a NOCA model with 8 hidden sources. Each source is associated with an 8×8 image pattern. The patterns and examples of the convoluted input images are shown in Figure 4, panels (a) and (b).

2. Standard regularization framework involves a data-independent term that penalizes for non-zero parameters. However, here the penalty term depends on data and is a property of the model.



(a) (b) (c)

Figure 5: Examples of models learned from 50, 200 and 1000 samples (panels a through c). The differences among models illustrate the improvement in the model recovery with increasing sample size. Although some source images are identified quite well with as few as 50 samples, the noise in other images is apparent. Models learned from 200 and 1000 samples are visibly improved.

4.1 Model Reconstruction

Our first objective is to assess the ability of the variational algorithm to learn the NOCA model from observational data. In this experiment, we used datasets of size 50 – 5000 datapoints that were generated randomly from the model. The datasets were given to the learning algorithm and the learned models were compared to the original model.

Figure 5 visualizes the parameters of three models recovered by the learning algorithm for varied sample sizes. It is apparent that the increase in the number of samples leads to improved models that are closer to the original model. The model learned from 50 samples suffers from high variance caused by the low number of training examples. Nevertheless, it is still able to capture some of the original source patterns. Sample sizes of 200 and 1000 improve the pattern recovery. By learning from 1000 samples, we were able to recover almost all sources used to generate data with a relatively small distortion.

Figure 5 illustrates the dependency of the model quality on the sample size in qualitative terms. To measure this dependency more rigorously we use the training/testing validation framework and a metric based on the joint distribution of observable data. The NOCA model is always learned from a training set. We use training sets of size $50, 100, 200, 500, 1000, 2000, 5000$. The testing set (sample size 2000) is viewed as a sample from the true multivariate distribution. We calculate its log-likelihood with respect to the learned model. A better fit of the model will be reflected in improved log-likelihood of the test sample with respect to this model. Figure 6 shows the log-likelihoods for NOCA models averaged over 50 testing sets. The results demonstrate that an increased size of training sets leads to a better log-likelihood of test data and hence a better approximation of the true distribution.

4.2 Model Selection

In practice, the correct latent dimensionality is rarely known in advance. Model selection is typically addressed within the Bayesian framework. Marginal data likelihood (Cooper and Herskovits, 1992) or its approximations (such as the Laplace approximation) are typically used for this purpose.

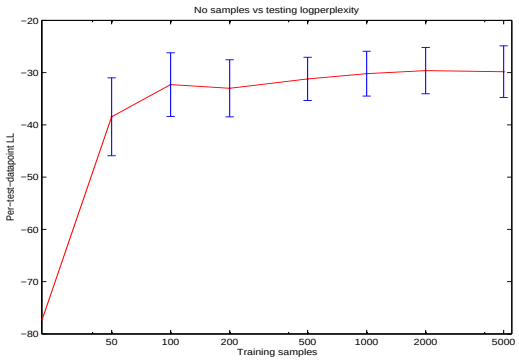


Figure 6: Average log-likelihoods of NOCA models on testing sets. The models are learned from training sets of size 50, 100, 200, 500, 1000, 2000 and 5000. The averages are over 50 trials. One-standard-deviation error bars are shown. The increase in the log-likelihood illustrates the improvement in the model recovery with an increasing sample size.

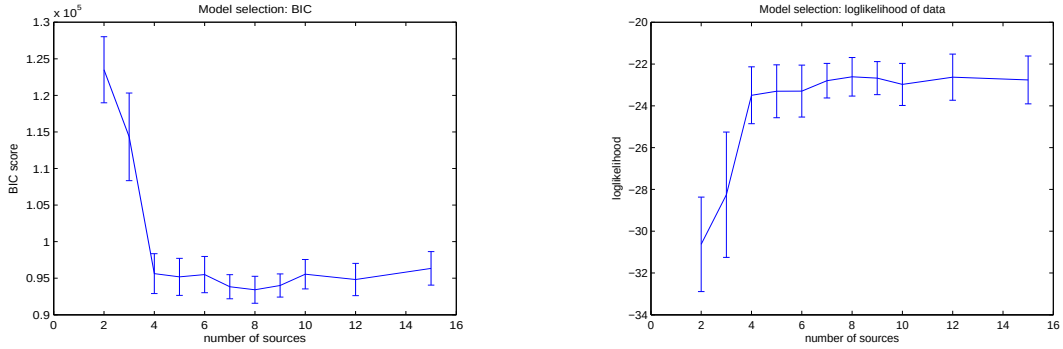
However, in presence of hidden variables it is intractable to compute the marginal likelihood. To address the model selection problem in NOCA we rely on the Bayesian Information Criterion (BIC). The BIC is a large-sample approximation to the integrated likelihood (Schwarz, 1978):

$$BIC(k) = -\ln p(\mathcal{D}|k,\hat{\Theta}_k) + \frac{1}{2}\psi_k \ln N$$

where $\hat{\Theta}_k$ is the ML estimate of NOCA parameters for the model with k hidden sources and ψ_k is the number of free parameters in this model.

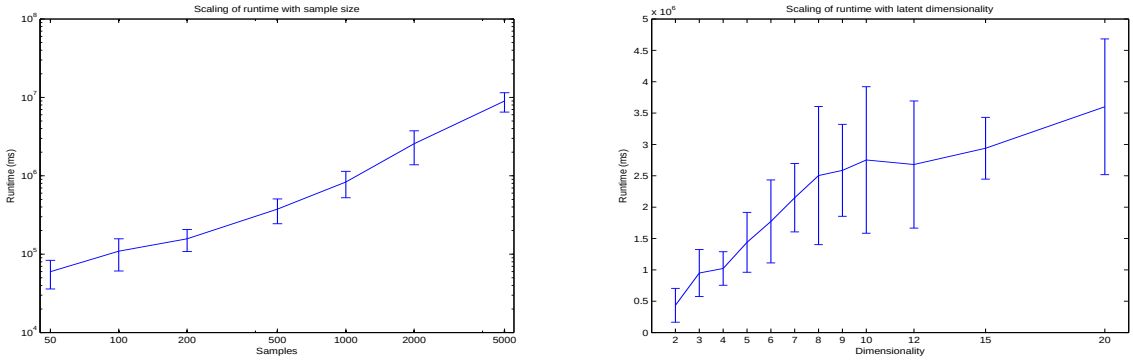
Figure 7a shows the results of model selection experiments based on the BIC score. The results are averages of BIC scores obtained by learning the model using 2000 images generated by sampling from NOCA model with 8 hidden sources. In training on this dataset, the number of assumed hidden sources varied from 2 to 15. To assure fair comparison, the same training data was used for all models in one train/test run. We see that the optimum BIC score is achieved at 8 sources which corresponds to number of sources in the original model.

The BIC score penalizes models with larger number of parameters. The penalty opposes the increase in the log-likelihood of training data we expect to see in more complex models with a larger number of hidden sources. However, in Section 3.3 we have pointed out the existence of an inherent "regularization" ability of NOCA, that is, its ability to shut down the influence of unnecessary sources once the true dimensionality of the model is reached. In such a case we would expect the log-likelihood of training data to level out for larger than the true number of sources. Figure 7b illustrates this point by plotting the log-likelihood of data for models with different number of sources. The setup of the experiment is the same as used in the BIC experiments. The log-likelihood score increases for models with fewer than 8 sources. The log-likelihood for more than 8 sources remains approximately the same. Visual inspection of the learned loading matrices reveals how this happens: many sources are disconnected from the model when the model learns their corresponding loading matrix rows to be identically 0. The models that were initialized with more than 8 sources most often stabilized at 7-8 active (connected) sources. The fact that in some instances the number



(a) (b)

Figure 7: **(a)** The average BIC scores for the models with varied number of sources. **(b)** The average log-likelihood of data for model with varied number of sources. In both cases, the true number of sources K is fixed at 8. Averages are calculated from 20 trials (one-standard-deviation bars are shown). In each trial, the model was learned using 2000 data points. The BIC reaches its optimal value at, and log-likelihood levels at 8 sources, which corresponds to true number of sources.



(a) (b)

Figure 8: **(a)** Runtimes of NOCA as they scale with increasing size of the training set. K is fixed at 8. **(b)** Scale-up with the number of assumed latent sources, the dataset size is fixed at 2000.

of sources converged to 7 can be explained the ability of the leak factor to effectively model an additional source.

4.3 Running-time Analysis

Precise time-complexity analysis of the NOCA learning algorithm is impossible since both the expectation and maximization steps involve iterative procedures whose convergence properties are not

well understood. Moreover, these are embedded in the EM loop itself and while eventual convergence is assured, its rate is not. Therefore we evaluate the time complexity empirically, with respect to N, the size of the training set and K, the number of latent sources.[3] We have observed no dependence between training set size, the assumed number of latent sources and the number of EM iterations performed in experiments.

The running time of the learning algorithm for different training set sizes is shown in Figure 8(a). A nearly straight line indicates that the complexity grows polynomially with the number of samples. In fact, we have observed that the time complexity scales approximately linearly with the number of samples in the trainset. The analysis of running times for different number of sources in Figure 8(b) shows that these scale roughly linearly with the number of assumed latent sources. This gives empirical support for the efficiency of the variational EM approximation as compared to the exact EM algorithm.

4.4 Dimensionality Reduction and Data Compression with NOCA

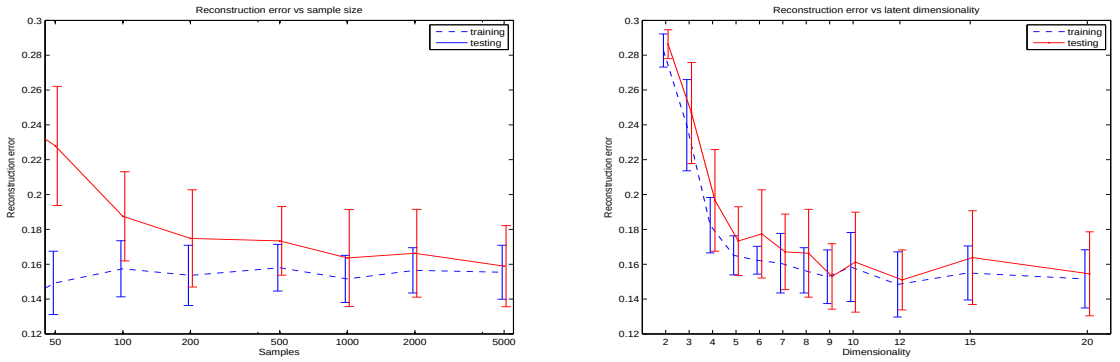
Latent variable models are inherently well suited for dimensionality reduction. Lossy compression of the data by the NOCA model can be achieved as follows. Given the learned NOCA model and an observed test-set image, we compute the posterior of each hidden source and pick the value with the higher posterior probability. The values of the hidden sources act as a low-dimensional representation of the test data. The high-dimensional data can then be recovered by sampling the observables given the stored values of sources and compared to the original test-set.

Figure 9(a) illustrates the data reconstruction error of the NOCA model learned for different sample sizes. The data reconstruction error is defined as the proportion of feature values in which the original dataset differs from the reconstructed data. We measure data reconstruction error on both the training and the testing data. The training set is the data used to learned the model, the testing set is an additional sample from the model. The data reconstruction error for the training set is smaller for very small sample sizes and stabilizes for sample sizes over 200. This can be explained by "overfitting" – the use of free model parameters for memorization of training data – for small sample sizes, and saturation of the model to its stochastic limit for larger sample sizes. The data reconstruction error for test sets behaves inversely – it is worse from smaller training sets and stabilizes for larger training sets as the learned model improves.

Figure 9(b) shows the influence of the number of hidden sources on the data reconstruction error. The data reconstruction error goes down with increasing K and flattens out as the learned models use no more than the true number of sources (8), thanks to the effect described in Section 3.3.

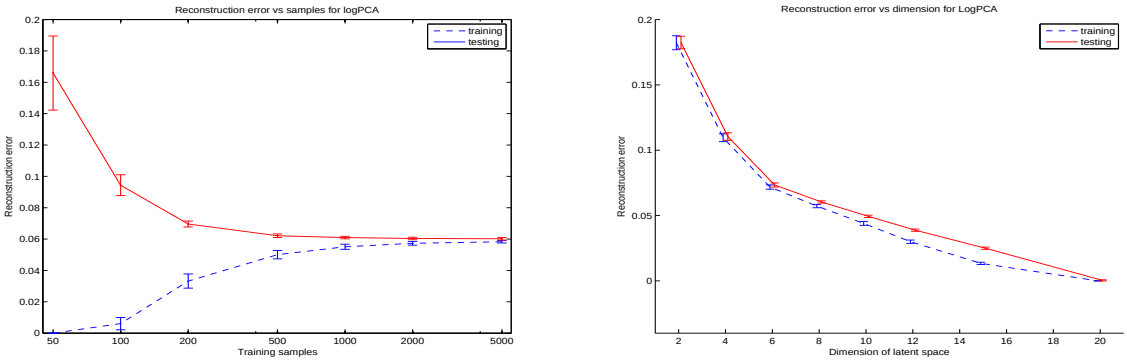
A related dimensionality-reduction model tailored to binary data is offered by logistic PCA (Schein et al., 2003). In this model, each component x_j^n of a datapoint $\mathbf{x}^n$ is assumed to be sampled from a Bernoulli distribution whose parameter θ_j^n is determined by a logistic unit from the factors $\mathbf{v}$, latent coordinates $\mathbf{u}$ and the bias term Δ_j: $\theta_j^n = \sigma(\mathrm{v}_j.\mathrm{u}_j^n + \Delta_j)$. The crucial difference between NOCA and logistic PCA is that the latent space in NOCA is discrete while in logistic PCA it is continuous. As a result, logistic PCA uses a many-bit floating point representation to capture many one-bit feature values. Figure 10 illustrates data reconstruction errors for the same experiments as performed for NOCA in Figure 9. The results demonstrate better data reconstruction performance of the logistic PCA model. This is expected since the complexity of NOCA's latent space is much

3. It follows from the form of the update equations that the algorithm is linear in D, the number of observable dimensions.



(a) (b)

Figure 9: **(a)** Average data reconstruction errors obtained for varied training sample sizes. **(b)** Average data reconstruction error plotted against the number of assumed latent sources. All values are averaged over 50 runs.



(a) (b)

Figure 10: Reconstruction errors achieved by the logistic PCA, a) as they vary with trainset size (fixed size testset), and b) as they vary with the latent dimension of the model.

smaller (*finite* as opposed to continuous). The differences in performance demonstrate the tradeoff in between the complexity of the representation of the latent space and its accuracy. In particular, NOCA uses 8 bits to represent each data point in the latent space while the logistic PCA uses a vector of 8 floating point values per data point.

5. An Application of NOCA to Citation Analysis

The analysis of NOCA on image datasets confirms it can discover, fully unsupervised, the structure of the hidden components reasonably well. But does the method apply to the real world? Do its assumptions really fit the data it was designed for? To assess this aspect of NOCA we test it on

a citation analysis problem. We first discuss the dataset and proceed to report the results of three evaluation strategies: (1) evaluation by a human judge, (2) a cosine-distance based metric and (3) perplexity of a testing set.

5.1 Citation Data

We acquired a dataset of approximately 17.000 documents from the CiteSeer online service. These are the HTML documents that place a scientific article within the lattice of citations; not to be mistaken for the actual text of the article. We chose forty authors active in these publication areas: *Introductions and tutorials*, *Markov chain Monte Carlo*, *Variational methods*, *Loopy belief propagation* and *Kernels and support vector machines*. Naturally, there are overlaps; for example, a publication discussing approximate inference in Bayesian networks is likely to mention both loopy belief propagation and MCMC techniques. This overlapping structure renders the task quite non-trivial. In addition, it makes it difficult to come up with an unambiguous "gold standard" clustering.

We selected all papers in the dataset citing any of the selected authors. The dataset was preprocessed into a binary matrix M_{ij}, where the element (i, j) is 1 if document i cites a paper authored by author j and 0 otherwise. Zero rows, that is documents that cite none of the authors, are discarded. There were 6592 non-zero rows in the matrix.

5.2 NOCA Formulation of Citation Analysis

The citation dataset consists of N documents, each of which cites a number of authors. The individual authors publish on one or more topics. Our conjecture is that certain citation patterns are indicative of paper topics. We wish to discover these topics and their associated authors, in a fully unsupervised manner.

To analyze the data with NOCA, we assume that the topics are represented with the hidden binary variables $s_1, \ldots s_K \in \{0,1\}$. Intuitively, $s_i = 1$ in the unobserved event that the document discusses topic i. The citation features correspond to the observed variables $x_1, \ldots x_D$. The n-th document in the corpus is thus represented by a D-dimensional binary vector $\mathbf{x}^n$. The event that document n cites author j is captured by observing $x_j^n = 1$. The "affinity" of author j and topic i is expressed by the weights p_{ij} which parameterize the noisy-or CPD's of the bottom layer nodes. This defines a generative probabilistic model at the document feature level:

- For all $i = 1, \ldots, K$, sample s_i from $Bernoulli(\pi_i)$
- For all $j = 1, \ldots, D$, sample x_j from the noisy-or distribution $p(x_j|\mathbf{s})$.

5.3 Mixture Models

Latent variable models have demonstrated good results in text and document analysis. Most of these are mixture models that view a document as a mixture of hidden topic factors. The topic factors are identified with distributions over words. The key assumption of a mixture model is that the occurrence of a specific word is determined by a *single* mixture component. These models share the bag-of-words view of a document and provide a probabilistic model for each word occurrence. NOCA offers a different view of a document: A document is a combination of *non-competing* topics and each word is determined by a *combination* of topics. NOCA does not define a model for generation of each single word, which makes it less suitable for applications such as text modeling, but it fits more naturally the type of data encountered in link analysis.

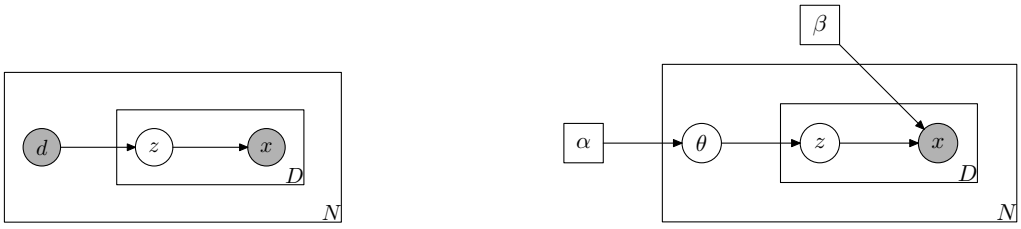


Figure 11: PLSA (a) and LDA (b) graphical models

In the following, we briefly review two mixture models applied frequently in text modeling: PLSA and LDA. These state-of-the-art text models have also been recast for link analysis purposes (Cohn and Hofmann, 2001; Cohn and Chang, 2000).

Probabilistic Latent Semantic Analysis (PLSA) (Hofmann, 1999a), whose graphical model is shown in Figure 11(a), assumes that each document is represented by a convex combination (a mixture) of topics and that the features of the document are generated by the following process:

1. pick a document d according to Multinomial $P(d)$ (defined by a dummy indexing of the documents in the dataset),
2. sample a topic z according to Multinomial $P(z|d)$,
3. generate a feature from $P(x|z)$.

The joint probability $P(d,x)$ factorizes as $P(d)\sum_z P(z|d)P(x|z)$. Since the topic variable z is unknown, the algorithm for learning PLSA derives from the EM framework.

Latent Dirichlet Allocation (LDA) (Blei et al., 2003) adds Bayesian hyperparameters to the PLSA model so that the mixture proportions themselves are a Dirichlet-distributed random variate (Figure 11). The following process is assumed to generate the documents:

1. sample a parameter θ from the exchangeable Dirichlet distribution $Dir(\alpha)$,
2. sample a topic from Multinomial $P(z|\theta)$,
3. generate a feature from $P(x|z,\beta)$.

Both the parameter θ and the topic variable z are unobserved. The addition of the new hidden parameters makes the exact inference for LDA intractable. To alleviate this problem Blei et al. derive a variational inference algorithm which in turn allows them to develop an efficient variational EM learning procedure.

The conceptual difference between NOCA on the one hand and PLSA or LDA on the other is that NOCA views a document as a *set of features*, while the mixture methods regard it as a *bag of words*. More importantly, NOCA makes a different assumption about the nature of the topic factors. PLSA (Figure 11, left) and LDA (Figure 11, right) view the topic factors as points in the vector space spanned by the orthogonal basis which is the vocabulary. Moreover, all these

points belong to a subspace of the $(D-1)$-dimensional word simplex since they correspond to normalized distributions. NOCA treats the topic as a separate type of entities that live in their own K-dimensional space which projects non-linearly into the vocabulary space. As opposed to PLSA, where one *aspect* is assumed to be responsible for the generation of a word, in NOCA, potentially all of the topic factors contribute to the generation of a single word feature. Additionally, the added freedom of the leak parameter allows NOCA to "put aside" the documents where no structure seems to stand out. These do not have to be accounted for in the output components. Clearer clustering is the outcome that we would expect from this organization.

5.4 Experiments

The evaluation of topic discovery in any of the frameworks relies on the identification of largest elements of output vectors or matrices. Since the semantics of the numeric values differs in the respective approaches, the only consistent way of comparing the outputs is by listing the most prominent elements of each of the identified clusters. We achieve this goal for different models as follows:

- Logistic PCA is parameterized by the loading matrix V and the constant bias vector Δ. We interpret rows of V as the component vectors and list the authors corresponding to the largest elements in each component vector.
- PLSA parameterization is not as in Figure 11(a), but instead the model is equivalently parameterized with $P(z)$, $P(d|z)$ and $P(x|z)$ (Hofmann, 1999a). We list the authors x with the highest $P(x|z)$ for each aspect z. Also reported is $P(z)$, to help assess the relative representation of the aspects.
- LDA provides the matrix β and the Dirichlet hyperparameter α. The reported components are the rows of β; the authors corresponding to the highest values in each row of β are listed. The components that LDA recovers are very stable, which is characteristic of the Bayesian approach taken in developing the model. Therefore we report results from 20 runs with different α (the initial exchangeable Dirichlet prior) instead, starting from $\alpha = 0.01$ and ending at $\alpha = 10$.
- For NOCA, the output consists of the cluster priors π_i, the loading matrix $\mathbf{p}$ and the "bias vector" $\mathbf{p}_0$. The authors listed under each component are those who received the highest weight in $\mathbf{p}_i$, the i-th row of the loading matrix $\mathbf{p}$. Again we report the priors π_i to compare the relative size of the clusters. Note that the priors need not sum to one, since each of them corresponds to a separate random variable.

5.4.1 QUALITATIVE EVALUATION

We ran all of the algorithms 20 times with different random initializations and visually judged the results from displays such as that in Figure 13. If a particular topic factor appeared and was determined to be of good "cluster purity", we assigned a score of 1 to the combination of community and analysis technique. If the cluster was identifiable with a community, but judged to be of mediocre purity, the score assigned was 1/2. Otherwise, the score assigned was 0. Whenever the community was captured in more than one factor, only one was counted. The maximum score is 20 as there were 20 experimental runs. The entries in Table 1 are the respective percentages.

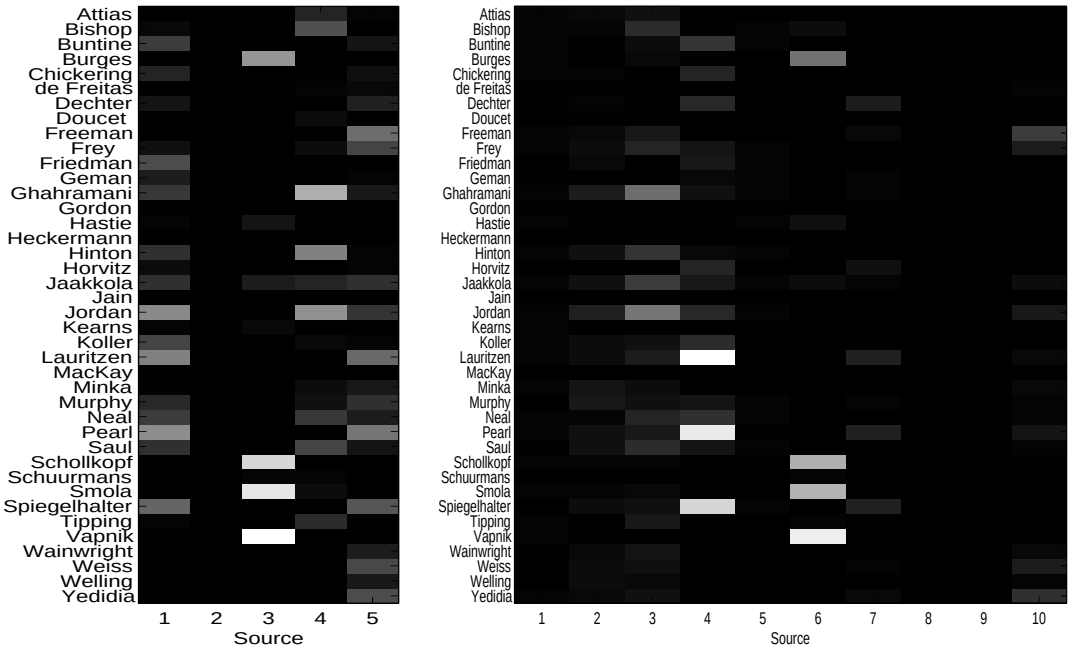


Figure 12: A result of noisy-or component analysis on the citation dataset. The columns visualize the parameters of the noisy-or loading matrix after they are rescaled by the prior of the source. Black fields correspond to 0s in the loading matrix, while white ones correspond to 1s.

(a) With 5 components. The following components are discernible:

- 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 contains many respected authors of basic references and tutorials on Bayesian belief networks.
- The second source was shut down 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. We can conclude that our algorithm found the LBP community.

(b) A run with 10 components illustrates the regularization behavior. Four out of ten sources were completely or almost completely shut off.

Comp. 1	Comp. 2	Comp. 3	Comp. 4	Comp. 5
Vapnik	Doucet	Bishop	Jain	Pearl
Doucet	de Freitas	Vapnik	Spiegelhalter	Smola
Freeman	Ghahramani	Hastie	Friedman	Lauritzen
Kearns	Friedman	Jain	Gordon	Friedman
Smola	Murphy	Burges	Bishop	Freeman
Murphy	Hastie	Schollkopf	Geman	Horvitz
de Freitas	Jordan	Smola	Neal	Schollkopf
?	**?**	**kernel**	**?**	**?**

(a) logPCA

Aspect 1	Aspect 2	Aspect 3	Aspect 4	Aspect 5
0.1284	0.1640	0.3513	0.2321	0.1241
Bishop	Friedman	Jordan	Vapnik	Geman
Jain	Neal	Hinton	Smola	Doucet
Kearns	Hastie	Spiegelhalter	Pearl	Gordon
Tipping	Murphy	Lauritzen	Schollkopf	de Freitas
Hinton	Koller	Pearl	Burges	Koller
Schuurmans	Jaakkola	Freeman	Horvitz	Frey
Saul	Buntine	Weiss	Koller	Murphy
?	**?**	**?**	**kernel**	**MCMC**

(b) PLSA

Comp. 1	Comp. 2	Comp. 3	Comp. 4	Comp. 5
0.0022	0.0912	0.0858	0.0277	0.0102
Minka	Vapnik	Lauritzen	Jordan	Freeman
Jordan	Smola	Pearl	Ghahramani	Yedidia
Jaakkola	Schollkopf	Spiegelhalter	Hinton	Weiss
Yedidia	Burges	Jordan	Bishop	Frey
Ghahramani	Hastie	Buntine	Saul	Murphy
Freeman	Jaakkola	Koller	Jaakkola	Welling
Frey	Bishop	Dechter	Attias	Pearl
?	**kernel**	**intro**	**variat'l**	**LBP**

(c) NOCA

Comp. 1	Comp. 2	Comp. 3	Comp. 4	Comp. 5
$\alpha = 1$				
Vapnik	Jordan	Geman	Friedman	Pearl
Smola	Hinton	Doucet	Koller	Lauritzen
Schollkopf	Neal	de Freitas	Hastie	Jain
Bishop	Ghahramani	Murphy	Kearns	Spiegelhalter
Burges	Weiss	Gordon	Buntine	Dechter
Tipping	Jaakkola	Koller	Chickering	Freeman
Jaakkola	Horvitz	Welling	Schuurmans	Frey
kernel	**variat'l**	**MCMC**	**intro**	**intro**

(d) LDA

Figure 13: Typical outputs from the link analysis algorithms:
a) Logistic PCA
b) Probabilistic latent semantic analysis. Also reported is the prior of each aspect $P(z = i)$.
c) Noisy-or component analysis. The prior on a source $P(s_i)$ is also shown.
d) Latent Dirichlet allocation with $\alpha = 1$.
Below each component, our evaluation of whether the component represents any of the the publication communities.

	Community				
Method	intro	MCMC	var'l	LBP	Kernel
LogPCA	40.0	42.5	15.0	10.0	67.5
PLSA	67.5	57.5	50.0	32.5	75.0
LDA	80.0	95.0	62.5	5.0	87.5
NOCA	85.0	15.0	92.5	82.5	75.0

Table 1: Success rates in recovering subcommunities in the citation data. The numbers are percentages averaged over 20 different random initializations.

The logistic PCA does not appear to be well suited for this task and is outperformed by the other methods. PLSA finds on average 2 communities in each run. LDA discovers the MCMC topic consistently, but fails to discover the LBP community. NOCA exhibits the opposite behavior: it reliably discovers LBP but fails to find the MCMC community most of the time. The SVM/kernel group and the variational methods community is consistently discovered by both NOCA and LDA, as well as the authors of widely cited overview and tutorial articles.

The difference observed for the LBP and MCMC communities is striking and should be explained by pointing out the characteristics of the respective communities. The LDA model is able to recover communities that have established their "market share" and have high enough prior probability that they are able to compete with the other groups for the direction that the topic simplex takes in the "vocabulary" space. LDA thus has a difficult time finding small, emerging areas. On the other hand, these nascent communities tend to be highly coherent, with a few pioneers that are very likely to be cited for their seminal papers. Such structure favors the NOCA model, which has a tendency to pick out tightly woven patterns and leave the more diffuse domains to be picked up by the leak factor. Thus the broader MCMC community eluded the noisy-or analyzer, while it was reliably captured by LDA; and the NOCA brought to light the LBP community.

In summary, NOCA discovers on average as many clusters as LDA, but the clusters are of different nature. If one wishes to gain insight into this type of data, we advocate that both methods be used, as they discover distinct kinds of patterns.

5.4.2 THE COSINE METRIC

While we took great care to assure objective evaluation, the above approach is nevertheless open to criticism on the grounds of "subjectivity." We would like our recovered components to align with a "gold standard," a set of vectors defined by a human *before* he or she sees the result of the clustering algorithm. Therefore we defined 0-1 vectors corresponding to the established communities as we perceive them. For example, the vector corresponding to LBP community has 1s at positions corresponding to names such as Freeman, Frey, Yedidia, etc. and 0s elsewhere.

A standard distance metric for vectors is their *cosine distance*. The similarity of two vectors $\mathbf{x}, \mathbf{y}$ is the cosine of their angle: $\cos\alpha(\mathbf{x},\mathbf{y}) = \frac{\mathbf{x}\cdot\mathbf{y}}{|\mathbf{x}||\mathbf{y}|}$. With the cosine metric, one can evaluate the similarity of two vectors. However, *how do we quantitatively evaluate a component set X as a whole?* Recovered components need to be matched to the original components as they can be permuted without affecting the likelihood. To obtain a one-to-one match, each original component is paired

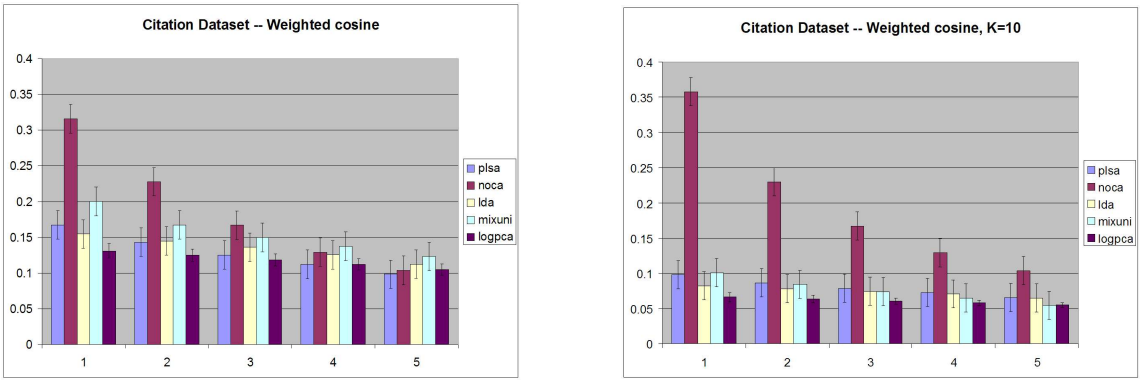


Figure 14: Weighted cosine similarities. On the horizontal axis is L, the number of components matched. The vertical axis shows the weighted cosine similarity. The left panel shows NOCA doing a superior job identifying the first few components, but it is soon overtaken by the mixture-based methods. The methods in the left panel operated with latent dimensionality 5, equal to the number of human-judged clusters. On the right, the picture changes when the latent dimensionality is increased to 10. While the performance of mixture-based methods deteriorates, NOCA's performance improves. This illustrates the difference in the assumptions about the data-generating process. The picture suggests that the more sophisticated methods do a better job in comparison with the baseline (a simple mixture of unigrams model) when the asumed latent dimensionality slightly exceeds the true number of clusters in the data.

with exactly one found by NOCA, so that the weighted sum of cosine distances is minimized. The weights u_i are defined so that they are proportional to the prior probabilities of the latent components and form a convex combination (sum to 1). The computation can be described by the formula

$$w_{\cos}(X,Y) = \min_{\pi,\rho} \sum_{i=1}^{K} u_{\pi(i)} \cos\alpha(\mathbf{x}_{\pi(i)}, \mathbf{y}_{\rho(i)}),$$

where π and ρ are permutations of the sets X and Y, respectively, and the minimization ranges over all possible permutations. Note that although this formula suggests evaluating exponentially many permutations, it effectively calls for finding a maximal-weight matching in a bipartite graph and can be computed efficiently. The resulting weighted cosine similarities are shown and commented in Figure 14.

5.4.3 PERPLEXITY

While the cosine scoring metric provides useful insights, using a standard probabilistic measure of model quality is in order to gauge how well the model estimates the joint density of the observable data. To assess this aspect of model recovery we rely on the cross-entropy of the "true" distribution and the distribution that the model entails. The testing set is viewed as a sample from the true multivariate distribution t and the *cross entropy* with the model distribution m is defined by $H(t,m) = -\sum_{\{\mathbf{x}\}} t(\mathbf{x}) \log m(\mathbf{x})$. Since the datapoints in the test set are by assumption inde-

	Cross-entropy			
K	NOCA	LDA	PLSA	MixUnigrams
5	$< 6.5 \pm 5.2$	$< 9.0 \pm 7.8$	6.1 ± 6.9	22.9 ± 31.3
10	$< 6.5 \pm 5.2$	$< 8.4 \pm 7.5$	4.9 ± 6.4	32.5 ± 46.0

Table 2: Cross-entropies between the model distribution and the empirical distribution induced by the test set. These numbers were obtained as mean and standard deviation on 20 train/test splits.

pendent and identically distributed, the cross entropy is approximately the average unconditional log-probability of datapoints in the test set (Cover and Thomas, 1991). *Perplexity* of the model m is defined as the quantity $2^{H(t,m)}$ and can be intuitively interpreted as the amount of information needed to predict the next datapoint. In short, the lower the cross-entropy is, the more precisely the model has learned the distribution of observables from the training set.

In the perplexity evaluation of NOCA and LDA, we use the tractable lower bounds on the document probabilities $P(\mathbf{x})$ (Equation 6 in this paper and Equation 13 in Blei et al. (2003)). The PLSA and logistic PCA models cannot be evaluated under the perplexity framework since they do not define a probability distribution on the test set. PLSA does define a distribution on the training set and the fold-in heuristic can be used (Hofmann, 1999b) so that it defines one on the test set. However, this heuristic gives PLSA an optical advantage over other models, as it allows it to refit the mixing proportions. As a baseline model, we will use a simple mixture of unigrams model. As NOCA provides no word-level model, but only a document-level probability model, we must compare all models in terms of document perplexity, instead of the standard approach that works at the level of words. Inspecting Table 2, we observe that the bound on perplexity of NOCA is significantly lower than that of LDA. PLSA shows a cross-entropy virtually on the level with NOCA, or slightly better. The cross-entropy of the baseline mixture-of-unigrams model is high, which is attributable to the data sparsity issue. Importantly, note that the values shown for LDA and NOCA are *lower bounds*, while the PLSA and MixUnigrams are exact.

6. Summary and conclusions

We have presented NOCA: a new latent-variable component analysis framework for high-dimensional binary data. To learn the NOCA model we have devised and presented an EM-based variational algorithm that overcomes the complexity limitation of exact learning methods. The proposed algorithm makes no assumption about the structure of the underlying noisy-or network, the structure is fully recovered during the learning process.

In addition to the component analysis task and related structure discovery problems, NOCA can be also used as a dimensionality reduction (data compression) tool, as well as a probabilistic model of high-dimensional binary data. We have tested these aspects of the model on a synthetic image decomposition problem and on a citation analysis problem of CiteSeer documents. The model and the algorithm showed favorable scale-up behavior and a very good model recovery and error reconstruction performance.

The task of community discovery has a natural formulation as a NOCA learning problem. A dataset of scientific paper citations in the field of machine learning was analyzed using the setup. The results, under several metrics, indicate that our algorithm performs on par with the current state-of-the-art mixture methods, but due to different data-generating assumptions it tends to expose different data structure. Such behavior is valuable as it enriches our insight into the intrinsic composition of the dataset.

Acknowledgments

We would like to thank the reviewers for their helpful comments and suggestions for improvements of the paper. This research was supported in part by the National Science Foundation grants ANI-0325353 and CMS-0416754 and by the University of Pittsburgh award CDRF-36851.

References

Hagai Attias. Independent Factor Analysis. *Neural Computation*, 11(4):803–851, 1999. URL citeseer.nj.nec.com/attias99independent.html.

David Bartholomew and Martin Knott. *Latent Variable Models and Factor Analysis*, volume 7 of *Kendall's Library of Statistics*. Oxford University Press, 1999.

Christopher Bishop. Latent variable models. In Michael Jordan, editor, *Learning in Graphical Models*, pages 371–403. MIT Press, 1999a.

Christopher Bishop. Variational principal components. In *Proceedings of* 9^{th} *International Conference on Artificial Neural Networks*, volume 1, pages 509–514, 1999b.

David Blei, Andrew Ng, and Michael Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, Jan 2003. URL http://www.cs.berkeley.edu/ blei/papers/blei03a.ps.gz.

Wray Buntine. Variational extensions to EM and multinomial PCA. In *Proceedings of the* 13^{th} *European Conference on Machine Learning*, 2002. URL citeseer.nj.nec.com/buntine02variational.html.

David Cohn and Huan Chang. Learning to probabilistically identify authoritative documents. In *Proceedings of the* 17^{th} *International Conference on Machine Learning*, pages 167–174. Morgan Kaufmann, San Francisco, CA, 2000. URL citeseer.ist.psu.edu/cohn00learning.html.

David Cohn and Thomas Hofmann. The missing link - a probabilistic model of document content and hypertext connectivity. In *Neural Information Processing Systems 13*, 2001. URL citeseer.ist.psu.edu/cohn01missing.html.

Gregory Cooper and Edward Herskovits. A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9:309–347, 1992.

Thomas Cover and Joy Thomas. *Elements of Information Theory*. John Wiley & sons, 1991.

Arthur Dempster, Nan Laird, and Donald Rubin. Maximum likelihood for incomplete data via the EM algorithm. *Journal of Royal Statistical Society*, 39:1–38, 1977.

Francisco Diez and Severino Gallan. Efficient computation for the Noisy-Max. *International Journal of Intelligent Systems*, 2003.

Zoubin Ghahramani and Michael Jordan. Factorial hidden Markov models. In David Touretzky, Michael Mozer, and Michael Hasselmo, editors, *Proceedings of Advances in Neural Information Processing Systems*, volume 8, pages 472–478. MIT Press, 1995. ISBN 0262201070. URL citeseer.nj.nec.com/article/ghahramani97factorial.html.

Zoubin Ghahramani and Michael Jordan. Factorial hidden Markov models. *Machine Learning*, 29: 245–273, 1997.

David Heckerman. Causal independence for knowledge acquisition and inference. In *Proceedings of 9th Conference on Uncertainty in AI UAI'93*, San Francisco, CA, 1993. Morgan Kaufmann Publishers.

Thomas Hofmann. Probabilistic latent semantic analysis. In *Proceedings of Uncertainty in Artificial Intelligence*, Stockholm, 1999a. URL citeseer.nj.nec.com/hofmann99probabilistic.html.

Thomas Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd Annual ACM Conference on Research and Development in Information Retrieval*, pages 50–57, Berkeley, California, August 1999b. URL citeseer.nj.nec.com/article/hofmann99probabilistic.html.

Tommi Jaakkola and Michael Jordan. Variational probabilistic inference and the QMR-DT network. *Journal of Artificial Intelligence Research*, 10:291–322, 1999. URL citeseer.nj.nec.com/article/jaakkola99variational.html.

Tommi Jaakkola, Lawrence Saul, and Michael Jordan. Fast learning by bounding likelihoods in sigmoid type belief networks. In David S. Touretzky, Michael C. Mozer, and Michael E. Hasselmo, editors, *Advances in Neural Information Processing Systems*, volume 8, pages 528–534. The MIT Press, 1996. URL citeseer.ist.psu.edu/jaakkola96fast.html.

Ian Jolliffe. *Principal Component Analysis*. Springer, 1986.

Michael Jordan, Zoubin Ghahramani, Tommi Jaakkola, and Lawrence Saul. An introduction to variational methods for graphical models. *Machine Learning*, 37(2):183–233, 1999.

Michael Kearns and Yishay Mansour. Exact inference of hidden structure from sample data in noisy-or networks. In *Proceedings of the* 14^{th} *Conference on Uncertainty in Artificial Intelligence*, pages 304–310, 1998. URL citeseer.nj.nec.com/383491.html.

Xinghua Lu, Milos Hauskrecht, and Roger Day. Modeling cellular processes with variational Bayesian cooperative vector quantizer. In *Proceedings of Pacific Symposium on Biocomputing*, 2004.

David MacKay. Probable networks and plausible predictions - a review of practical Baysian methods for supervised neural networks. *Network: Computation in Neural Systems*, 6(3):469–505, 1995.

James Miskin. *Ensemble Learning for Independent Component Analysis*. PhD thesis, Selwyn College, University of Cambridge, 2000.

David Ross and Richard Zemel. Multiple cause vector quantization. In *Proceedings of Advances in Neural Information Processing Systems 16*, 2002. URL citeseer.ist.psu.edu/ross02multiple.html.

Lawrence Saul, Tommi Jaakkola, and Michael Jordan. Mean field theory for sigmoid belief networks. *Journal of Artificial Intelligence Research*, 4:61–76, 1996.

Andrew Schein, Lawrence Saul, and Lyle Ungar. A generalized linear model for principal component analysis of binary data. In *Proceedings of the* 9^{th} *International Workshop on Artificial Intelligence and Statistics*, 2003. URL citeseer.nj.nec.com/546431.html.

Gideon Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6:461–464, 1978.

Michael Shwe, Blackford Middleton, David Heckerman, Max Henrion, Eric Horvitz, Harold Lehmann, and Gregory Cooper. Probabilistic diagnosis using a reformulation of the INTERNIST-1/QMR knowledge base: Part I. The probabilistic model and inference algorithms. *Methods of Information in Medicine*, 30:241–255, 1991.

Tomáš Šingliar and Miloš Hauskrecht. Variational learning for noisy-or component analysis. In *Proceedings of the SIAM International Conference on Data Mining*, pages 370–379, 2005.

Michael Tipping and Christopher Bishop. Probabilistic principal component analysis. Technical Report NCRG/97/010, Neural Computing Research Group, Aston University, September 1997. URL citeseer.nj.nec.com/article/tipping97probabilistic.html.

Jiří Vomlel. Noisy-or classifier. In *Proceedings of the* 6^{th} *Workshop on Uncertainty Processing*, pages 291–302, 2003. URL http://lisp.vse.cz/wupes2003/.