PUBLICATION 1

Ella Bingham and Aapo Hyvärinen. A fast fixed-point algorithm for independent component analysis of complex valued signals. *International Journal of Neural Systems*, 10(1):1–8, February 2000.

International Journal of Neural Systems, Vol. 10, No. 1 (February, 2000) 1–8

A FAST FIXED-POINT ALGORITHM FOR INDEPENDENT COMPONENT ANALYSIS OF COMPLEX VALUED SIGNALS

ELLA BINGHAM* and AAPO HYVÄRINEN†
Neural Networks Research Centre, Helsinki University of Technology,
P.O. Box 5400, FIN-02015 HUT, Finland
**E-mail*: *Ella.Bingham@hut.fi*
†*E-mail*: *Aapo.Hyvarinen@hut.fi*
http://www.cis.hut.fi/projects/ica/

Received 21 October 1999
Revised 26 January 2000
Accepted 26 January 2000

Separation of complex valued signals is a frequently arising problem in signal processing. For example, separation of convolutively mixed source signals involves computations on complex valued signals. In this article, it is assumed that the original, complex valued source signals are mutually statistically independent, and the problem is solved by the independent component analysis (ICA) model. ICA is a statistical method for transforming an observed multidimensional random vector into components that are mutually as independent as possible. In this article, a fast fixed-point type algorithm that is capable of separating complex valued, linearly mixed source signals is presented and its computational efficiency is shown by simulations. Also, the local consistency of the estimator given by the algorithm is proved.

1. Introduction

Separation of complex valued signals is a frequently arising problem in signal processing: frequency-domain implementations involving complex valued signals have advantages over time-domain implementations. Especially in the separation of convolutive mixtures it is a common practice to Fourier transform the signals, which results in complex valued signals. In this article, we present an algorithm for the separation of complex valued signals. Our framework is Independent Component Analysis.

Independent component analysis (ICA)[1,2] is a statistical model where the observed data is expressed as a linear combination of underlying latent variables. The latent variables are assumed non-Gaussian and mutually independent. The task is to find out both the latent variables and the mixing process. The ICA model used in this article is

$$\mathbf{x} = \mathbf{A}\mathbf{s} \tag{1}$$

where $\mathbf{x} = (x_1, \ldots, x_m)$ is the vector of observed random variables, $\mathbf{s} = (s_1, \ldots, s_n)$ is the vector of statistically independent latent variables called the independent components, and $\mathbf{A}$ is an unknown constant mixing matrix. The above model is identifiable under the following fundamental restrictions:[1] at most one of the independent components s_j may be Gaussian, and the matrix $\mathbf{A}$ must be of full column rank. (The identifiability of the model is proved in Ref. 1 in the case $n = m$.)

A fast fixed point algorithm (FastICA) for the separation of linearly mixed independent source signals was presented by Hyvärinen and Oja.[3,4] The FastICA algorithm is a computationally efficient and robust fixed-point type algorithm for independent component analysis and blind source separation.

In this article, we show how the FastICA algorithm can be extended to complex valued signals. Both the independent component variables $\mathbf{s}$ and the observed variables $\mathbf{x}$ in model (1) assume complex

values. For simplicity, the number of independent component variables is the same as the number of observed linear mixtures, that is, $n = m$. The mixing matrix $\mathbf{A}$ is of full rank and it may be complex as well, but this is optional. A necessary preprocessing of the data $\mathbf{x}$ is whitening, which can always be accomplished by e.g., Principal Component Analysis.[1] We assume that the signals s_j are zero-mean and white, i.e., real and imaginary parts of s_j are uncorrelated and their variances are equal; this is quite realistic in practical problems.

Algorithms for independent component analysis of complex valued signals are also presented in Refs. 5 and 6. Both of these algorithms are computationally more intensive than our algorithm, and no proofs of consistency are given in either of the references. In contrast, we prove the local consistency of the estimator given by our algorithm, and show its computational efficiency by simulations. Our algorithm is also more robust against outliers than kurtosis-based ICA algorithms (see Ref. 3 for a discussion on robust estimators for ICA). Also, our algorithm is capable of deflationary separation of the independent component signals; it is possible to estimate only one or some of the independent components, which is useful if the exact number of independent components is not known beforehand. In deflationary separation the components tend to separate in the order of decreasing non-Gaussianity, which often equals decreasing "importance" of the components.

This paper is organized as follows. We first go through some basic concepts of complex random variables in Sec. 2. We then discuss the indeterminacy that is inherent in estimating complex valued independent components (Sec. 3). In Sec. 4, we motivate our approach of ICA estimation and discuss the contrast function used in our algorithm. The fast fixed-point algorithm is presented in Sec. 5, and simulation results confirming the usefulness of the algorithm are shown in Sec. 6. Section 7 discusses connections to other ICA research. Finally, some conclusions are drawn in Sec. 8.

2. Basic Concepts of Complex Random Variables

A complex random variable may be represented as $y = u + iv$ where u and v are real-valued random variables. The density of y is $f(y) = f(u, v) \in \mathbb{R}^2$. The *expectation* of y is $E\{y\} = E\{u\} + iE\{v\}$. Two complex random variables y_1 and y_2 are *uncorrelated* if $E\{y_1 y_2^*\} = E\{y_1\}E\{y_2^*\}$, where y^* designates the complex conjugate of y. The *covariance matrix* of a zero-mean complex random vector $\mathbf{y} = (y_1, \ldots, y_n)$ is

$$E\{\mathbf{y}\mathbf{y}^H\} = \begin{bmatrix} C_{11} & \cdots & C_{1n} \\ \vdots & \ddots & \vdots \\ C_{n1} & \cdots & C_{nn} \end{bmatrix} \tag{2}$$

where $C_{jk} = E\{y_j y_k^*\}$ and $\mathbf{y}^H$ stands for the Hermitian of $\mathbf{y}$, that is, $\mathbf{y}$ transposed and conjugated. In our complex ICA model, all source signals s_j are zero-mean and they have unit variances and uncorrelated real and imaginary parts of equal variances. In short, these requirements are equivalent to $E\{\mathbf{s}\mathbf{s}^H\} = \mathbf{I}$ and $E\{\mathbf{s}\mathbf{s}^T\} = \mathbf{O}$. In the latter, the expectation of the outer product of a *complex* random vector *without* the conjugate is a null matrix. These assumptions imply that s_j must be strictly complex; that is, the imaginary part of s_j may not in general vanish.

A frequently encountered statistics in ICA is *kurtosis*, or fourth-order cumulant. For zero-mean, complex random variables it could be defined, for example, as in Refs. 6 and 7

$$\begin{aligned} \mathrm{kurt}(y) = {} & E\{|y|^4\} - E\{yy^*\}E\{yy^*\} - E\{yy\}E\{y^*y^*\} \\ & - E\{yy^*\}E\{y^*y\} \end{aligned} \tag{3}$$

but the definitions vary with respect to the placement of conjugates (*) — actually, there are 2^4 ways to define the kurtosis.[7] We choose the definition in Ref. 8, where

$$\begin{aligned} \mathrm{kurt}(y) &= E\{|y|^4\} - 2(E\{|y|^2\})^2 - |E\{y^2\}|^2 \\ &= E\{|y|^4\} - 2 \end{aligned} \tag{4}$$

where y is white, i.e., the real and imaginary parts of y are uncorrelated and their variances are equal. This definition of kurtosis is intuitive since it vanishes if y is Gaussian.

3. Indeterminacy of the Independent Components

The independent components $\mathbf{s}$ in the ICA model (1) are found by searching for a matrix $\mathbf{W}$ such that

$\mathbf{s} = \mathbf{W}^H\mathbf{x}$ up to some indeterminacies, which are discussed in the following. In this paper, we use the notation $\mathbf{s} = \mathbf{W}^H\mathbf{x}$ which is analogous to the notation in Ref. 4 but differs from the notation $\mathbf{s} = \mathbf{W}\mathbf{x}$ used in Ref. 3.

In the real case, a scalar factor $\alpha_j \in \mathbb{R}$, $\alpha_j \neq 0$ can be exchanged between s_j and a column $\mathbf{a}_j$ of $\mathbf{A}$ without changing the distribution of $\mathbf{x}$: $\mathbf{a}_j s_j = (\alpha_j \mathbf{a}_j)(\alpha_j^{-1} s_j)$. In other words, the order, the signs and the scaling of the independent components cannot be determined. Anyhow, the order of s_j may be chosen arbitrarily and it is a common practice to set $E\{s_j^2\} = 1$; thus only the signs of the independent components are indetermined.

Similarly in the complex case there is an unknown phase v_j for each s_j: it is easily proved that

$$\mathbf{a}_j s_j = (v_j \mathbf{a}_j)\left(\frac{s_j}{v_j}\right), \quad |v_j| = 1,\, v_j \in \mathbb{C}\,. \tag{5}$$

If s_j has a spherically symmetric distribution, i.e., the distribution depends on the modulus of s_j only, the multiplication by a variable v_j does not change the distribution of s_j. Thus the distribution of $\mathbf{x}$ remains unchanged as well.

From this indeterminacy it follows that it is impossible to retain the phases of s_j, and $\mathbf{W}^H\mathbf{A}$ is a matrix where in each row and each column there is one nonzero element $v_j \in \mathbb{C}$ that is of unit modulus. Note that the indeterminacy is an inherent property of complex ICA — it does not follow from the assumptions made in this article.

4. Contrast Function

4.1. *Choice of the contrast function*

Now we generalize the framework in Refs. 3, 4 and 9 for complex valued signals. One might make a distinction between "top-down" and "bottom-up" approaches to ICA.[9] In the top-down approach, independence is measured by such measures as mutual information which is often approximated by using cumulants. This may result in non-robust contrast functions and burdensome computations. We choose here the bottom-up approach, where the higher-order statistics are implicitly embedded into the algorithm by arbitrary non-linearities. We start from an arbitrary non-linear contrast function and prove that its extrema coincide with the independent components. This bottom-up approach is computationally simple, and the non-linearity can be chosen quite freely to optimize e.g., the statistical behavior of the estimator.

Our contrast function is

$$J_G(\mathbf{w}) = E\{G(|\mathbf{w}^H\mathbf{x}|^2)\} \tag{6}$$

where $G : \mathbb{R}^+ \cup \{0\} \to \mathbb{R}$ is a smooth even function, $\mathbf{w}$ is an n-dimensional complex weight vector and $E\{|\mathbf{w}^H\mathbf{x}|^2\} = 1$. Finding the extrema of a contrast function is a well defined problem only if the function is real. For this reason our contrast functions operate on absolute values rather than on complex values.

Remember Formula (4) for the kurtosis of complex variables: if we choose $G(y) = y^2$, then $J_G(\mathbf{w}) = E\{|\mathbf{w}^H\mathbf{x}|^4\}$. Thus J essentially measures the kurtosis of $\mathbf{w}^H\mathbf{x}$, which is a classic measure in higher-order statistics.

Maximizing the sum of n one-unit contrast functions, and taking into account the constraint of decorrelation, one obtains the following optimization problem:

$$\begin{aligned} &\text{maximize } \sum_{j=1}^{n} J_G(\mathbf{w}_j) \text{ with respect to } \mathbf{w}_j, \\ &\qquad\qquad\qquad\qquad j = 1, \ldots, n \\ &\text{under constraint } E\{(\mathbf{w}_k^H\mathbf{x})(\mathbf{w}_j^H\mathbf{x})^*\} = \delta_{jk} \end{aligned} \tag{7}$$

where $\delta_{jk} = 1$ for $j = k$ and $\delta_{jk} = 0$ otherwise.

It is highly preferable that the estimator given by the contrast function is robust against outliers. The more slowly G grows as its argument increases, the more robust is the estimator. For the choice of G we propose now three different functions, the derivatives g of which are also given:

$$G_1(y) = \sqrt{a_1 + y}\,, \quad g_1(y) = \frac{1}{2\sqrt{a_1 + y}} \tag{8}$$

$$G_2(y) = \log(a_2 + y)\,, \quad g_2(y) = \frac{1}{a_2 + y} \tag{9}$$

$$G_3(y) = \frac{1}{2}\,y^2\,, \quad g_3(y) = y \tag{10}$$

where a_1 and a_2 are some arbitrary constants for which values $a_1 \approx 0.1$ and $a_2 \approx 0.1$ were chosen in this work. Of the above functions, G_1 and G_2 grow more slowly than G_3 and thus they give more robust estimators. G_3 is motivated by kurtosis (4).

4.2. *Consistency*

In Ref. 9, in the context of ICA on real-valued signals, it was stated that any non-linear learning function G divides the space of probability distributions into two half-spaces. Independent components can be estimated by either maximizing or minimizing a function similar to (6), depending on which half-space their distribution lies in. In Ref. 9, a theorem for real valued signals was presented that distinguished between maximization and minimization and gave the exact conditions for convergence. In the following, we show how this idea can be generalized to complex valued random variables. We have the following theorem on the local consistency of the estimators, the proof of which is given in the Appendix:

Theorem

Assume that the input data follows the model (1). The observed variables x_k, $k = 1, \ldots, n$ in $\mathbf{x}$ are prewhitened using $E\{\mathbf{x}\mathbf{x}^H\} = \mathbf{I}$. The independent component variables s_k, $k = 1, \ldots, n$ in $\mathbf{s}$ are zero-mean and have unit variances and uncorrelated real and imaginary parts of equal variances. Also, $G : \mathbb{R}^+ \cup \{0\} \to \mathbb{R}$ is a sufficiently smooth even function. Then the local maxima (resp. minima) of $E\{G(|\mathbf{w}^H\mathbf{x}|^2)\}$ under the constraint $E\{|\mathbf{w}^H\mathbf{x}|^2\} = \|\mathbf{w}\|^2 = 1$ include those rows $\mathbf{a}_k$ of the inverse of the mixing matrix $\mathbf{A}$ such that the corresponding independent components s_k satisfy

$$E\{g(|s_k|^2) + |s_k|^2 g'(|s_k|^2) - |s_k|^2 g(|s_k|^2)\} < 0 \quad (> 0, \text{ resp.}) \tag{11}$$

where $g()$ is the derivative of $G()$ and $g'()$ is the derivative of $g()$. The same is true for the points $-\mathbf{a}_k$.

A special case of the theorem is when $g(y) = y$, $g'(y) = 1$. Condition (11) reads now

$$E\{|s_k|^2 + |s_k|^2 - |s_k|^2|s_k|^2\} = -E\{|s_k|^4\} + 2 < 0 \quad (> 0, \text{ resp.}). \tag{12}$$

Thus the local maxima of $E\{G(|\mathbf{w}^H\mathbf{x}|^2)\}$ are found when $E\{|s_k|^4\} - 2 > 0$, i.e., the kurtosis (4) of s_k is positive.

5. Fixed-Point Algorithm

We now give the fixed-point algorithm for complex signals under the ICA data model (1). The algorithm searches for the extrema of $E\{G(|\mathbf{w}^H\mathbf{x}|^2)\}$. Details of the derivation are presented in the Appendix.

The algorithm requires a preliminary sphering or whitening of the data: the observed variable $\mathbf{x}_{\text{old}}$ is linearly transformed to a zero-mean variable $\mathbf{x} = \mathbf{Q}\mathbf{x}_{\text{old}}$, $\mathbf{x} = (x_{1r} + ix_{1i}, \ldots, x_{nr} + ix_{ni})$ such that $E\{\mathbf{x}\mathbf{x}^H\} = \mathbf{I}$. Whitening can always be accomplished by e.g., Principal Component Analysis.[1]

The fixed-point algorithm for one unit is

$$\begin{aligned} \mathbf{w}^+ &= E\{\mathbf{x}(\mathbf{w}^H\mathbf{x})^* g(|\mathbf{w}^H\mathbf{x}|^2)\} - E\{g(|\mathbf{w}^H\mathbf{x}|^2) \\ &\quad + |\mathbf{w}^H\mathbf{x}|^2 g'(|\mathbf{w}^H\mathbf{x}|^2)\}\mathbf{w} \\ \mathbf{w}_{\text{new}} &= \frac{\mathbf{w}^+}{\|\mathbf{w}^+\|}. \end{aligned} \tag{13}$$

The one-unit algorithm can be extended to the estimation of the whole ICA transformation $\mathbf{s} = \mathbf{W}^H\mathbf{x}$. To prevent different neurons from converging to the same maxima, the outputs $\mathbf{w}_1^H\mathbf{x}, \ldots, \mathbf{w}_n^H\mathbf{x}$ are decorrelated after every iteration. A simple way to accomplish this is a deflation scheme based on a Gram-Schmidt-like decorrelation:[3] When we have estimated p independent components, or p vectors $\mathbf{w}_1, \ldots, \mathbf{w}_p$, we run the one-unit fixed-point algorithm for $\mathbf{w}_{p+1}$, and after every iteration step subtract from $\mathbf{w}_{p+1}$ the projections of the previously estimated p vectors, and then renormalize $\mathbf{w}_{p+1}$:

$$\begin{aligned} \mathbf{w}_{p+1} &= \mathbf{w}_{p+1} - \sum_{j=1}^{p} \mathbf{w}_j \mathbf{w}_j^H \mathbf{w}_{p+1} \\ \mathbf{w}_{p+1} &= \frac{\mathbf{w}_{p+1}}{\|\mathbf{w}_{p+1}\|}. \end{aligned} \tag{14}$$

The above decorrelation scheme is suitable for deflationary separation of the independent components. Sometimes it is preferable to estimate all the independent components simultaneously, and use a symmetric decorrelation. This can be accomplished e.g., by

$$\mathbf{W} = \mathbf{W}(\mathbf{W}^H\mathbf{W})^{-1/2} \tag{15}$$

where $\mathbf{W} = (\mathbf{w}_1 \cdots \mathbf{w}_n)$ is the matrix of the vectors.

6. Simulation Results

Complex signals were separated to test the performance of the fast fixed-point algorithm and the Theorem. Symmetric decorrelation scheme, presented in Formula (15), was used in the algorithm. The data were artificially generated complex random signals $s_j = r_j(\cos\,\phi_j + i\,\sin\,\phi_j)$ where for each signal j the radius r_j was drawn from a different distribution and the phase angle ϕ_j was uniformly distributed on $[-\pi, \pi]$, which implied that real and imaginary parts of the signals were uncorrelated and of equal variance. These assumptions are quite realistic in practical problems. Also, each signal was normalized to unit variance. There were a total of eight complex random signals and 50,000 samples per signal at each trial.

Source signals $\mathbf{s}$ were mixed using a randomly generated complex mixing matrix $\mathbf{A}$. The mixed signals $\mathbf{x}_{\rm old} = \mathbf{A}\mathbf{s}$ were first whitened using $\mathbf{x} = \mathbf{Q}\mathbf{x}_{\rm old}$ and then fed to the fixed point algorithm. A complex unmixing matrix $\mathbf{W}$ was sought so that $\mathbf{s} = \mathbf{W}^H\mathbf{x}$. The result of the separation can be measured by $\mathbf{W}^H(\mathbf{Q}\mathbf{A})$. It should converge to a matrix where in each row and each column there is one non-zero element $v \in \mathbb{C}$ of unit modulus; i.e., in the end, $|\mathbf{W}^H(\mathbf{Q}\mathbf{A})|$ should be a permutation matrix. Our error measure is the sum of squared deviation of $|\mathbf{W}^H(\mathbf{Q}\mathbf{A})|$ from the nearest permutation matrix.

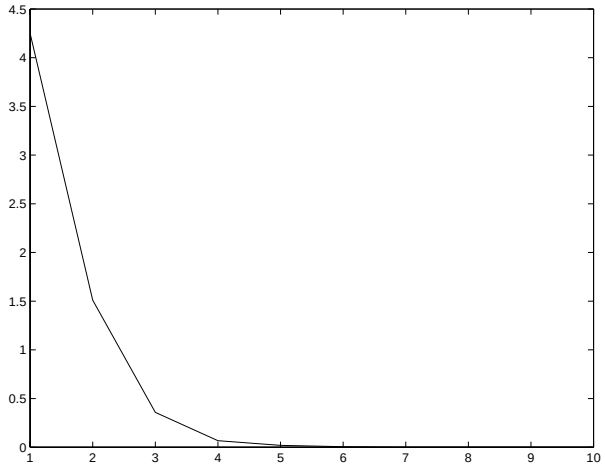

Fig. 1. Convergence of the fixed-point algorithm using contrast function $G_2(y) = \log(a_2 + y)$; average result over ten runs. About six iteration steps were needed for convergence.

All three contrast functions were successful in that the Theorem was always fulfilled and $|\mathbf{W}^H(\mathbf{Q}\mathbf{A})|$ converged to a permutation matrix in about six steps. Figure 1 shows the convergence using G_2.

7. Relation to Subspace Methods

Our complex ICA closely resembles independent subspace methods[10] and multidimensional ICA.[11] In both methods, the components s_j can be divided into m-tuples such that the components inside a given m-tuple may be dependent on each other but independent of other m-tuples. Each m-tuple corresponds to m basis vectors that are orthogonal after prewhitening. In Ref. 10, it was proposed that the distributions inside the m-tuples could be modeled by spherically symmetric distributions. This implies that the contrast function (for one subspace) should be of the form $E\{G(\sum_{j=1}^{m}(\mathbf{w}_j^T\mathbf{x})^2)\}$ where $\mathbf{w}_j^T\mathbf{w}_k = 0$, $j \neq k$.

In our complex ICA, the contrast function operates on $|\mathbf{w}^H\mathbf{x}|^2$ which may be expressed as $(\tilde{\mathbf{w}}^T\tilde{\mathbf{x}})^2 + (\tilde{\mathbf{w}}'^T\tilde{\mathbf{x}})^2$. Here $\mathbf{w} = (w_{1r} + iw_{1i}, \ldots, w_{nr} + iw_{ni})$, $\mathbf{x} = (x_{1r} + ix_{1i}, \ldots, x_{nr} + ix_{ni})$, $\tilde{\mathbf{w}} = (w_{1r}, w_{1i}, \ldots, w_{nr}, w_{ni})$, $\tilde{\mathbf{w}}' = (-w_{1i}, w_{1r}, \ldots, -w_{ni}, w_{nr})$ and $\tilde{\mathbf{x}} = (x_{1r}, x_{1i}, \ldots, x_{nr}, x_{ni})$. Thus the subspace is two-dimensional (real and imaginary parts of a complex number) and there are two orthogonal basis vectors: $\tilde{\mathbf{w}}^T\tilde{\mathbf{w}}' = 0$. In contrast to subspace methods, one of the basis vectors is determined straightforward from the other basis vector.

In independent subspace analysis, the independent subspace is determined only up to an orthogonal $m \times m$ matrix factor.[10] In complex ICA however, the indeterminacy is less severe: the sources are determined up to a complex factor v, $|v| = 1$.

It can be concluded that complex ICA is a restricted form of independent subspace methods.

8. Conclusion

We have presented a fixed-point type algorithm for the separation of linearly mixed, complex valued signals in the ICA framework. Our algorithm is based on a deflationary separation of independent components. The algorithm is robust against outliers and computationally simple, and the estimator given by the algorithm is locally consistent. We have also shown the computational efficiency of the algorithm by simulations.

Appendix A

Proof of Theorem

Denote by $H(\mathbf{w})$ the function to be minimized or maximized, $E\{G(|\mathbf{w}^H\mathbf{x}|^2)\}$. Make the orthogonal change of coordinates $\mathbf{z} = \mathbf{A}^H\mathbf{w}$, giving $H(\mathbf{z}) = E\{G(|\mathbf{z}^H\mathbf{s}|^2)\}$. When $\mathbf{w}$ coincides with one of the rows of $\mathbf{A}^{-1}$, we have $\mathbf{z} = (0, \ldots, 0, v, 0, \ldots, 0)$ — remember that $\mathbf{A}$ is orthogonal due to the prewhitening of $\mathbf{x}$. In the following, we shall analyze the stability of such $\mathbf{z}$.

We now search for a Taylor expansion of H in the extrema. We do not use complex differentiation operators because H is in general not analytic and thus it cannot be expanded as a Taylor series in the complex form. The gradient of H with respect to $\mathbf{z}$ is

$$\begin{aligned}\nabla H(\mathbf{z}) &= \begin{pmatrix} \frac{\partial}{\partial z_{1r}} \\ \frac{\partial}{\partial z_{1i}} \\ \vdots \\ \frac{\partial}{\partial z_{nr}} \\ \frac{\partial}{\partial z_{ni}} \end{pmatrix} H(\mathbf{z}) \\ &= 2 \begin{pmatrix} E\{\mathrm{Re}\{s_1(\mathbf{z}^H\mathbf{s})^*\}\, g(|\mathbf{z}^H\mathbf{s}|^2)\} \\ E\{\mathrm{Im}\{s_1(\mathbf{z}^H\mathbf{s})^*\}\, g(|\mathbf{z}^H\mathbf{s}|^2)\} \\ \vdots \\ E\{\mathrm{Re}\{s_n(\mathbf{z}^H\mathbf{s})^*\}\, g(|\mathbf{z}^H\mathbf{s}|^2)\} \\ E\{\mathrm{Im}\{s_n(\mathbf{z}^H\mathbf{s})^*\}\, g(|\mathbf{z}^H\mathbf{s}|^2)\} \end{pmatrix}\end{aligned} \tag{16}$$

where $z_j = z_{jr} + iz_{ji}$ and $s_j = s_{jr} + is_{ji}$.

The Hessian of H is now a $2n \times 2n$ real matrix: denote ∇H as $(h_{R1}, h_{I1}, \ldots, h_{Rn}, h_{In})$ where

$$h_{Rj} = E\{\mathrm{Re}\{s_j(\mathbf{z}^H\mathbf{s})^*\}\, g(|\mathbf{z}^H\mathbf{s}|^2)\} \tag{17}$$

$$h_{Ij} = E\{\mathrm{Im}\{s_j(\mathbf{z}^H\mathbf{s})^*\}\, g(|\mathbf{z}^H\mathbf{s}|^2)\} \tag{18}$$

whence the Hessian of H is

$$\nabla^2 H(\mathbf{z}) = 2 \begin{pmatrix} \frac{\partial h_{R1}}{\partial z_{1r}} & \frac{\partial h_{R1}}{\partial z_{1i}} & \cdots & \frac{\partial h_{R1}}{\partial z_{nr}} & \frac{\partial h_{R1}}{\partial z_{ni}} \\ \frac{\partial h_{I1}}{\partial z_{1r}} & \frac{\partial h_{I1}}{\partial z_{1i}} & \cdots & \frac{\partial h_{I1}}{\partial z_{nr}} & \frac{\partial h_{I1}}{\partial z_{ni}} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \frac{\partial h_{Rn}}{\partial z_{1r}} & \frac{\partial h_{Rn}}{\partial z_{1i}} & \cdots & \frac{\partial h_{Rn}}{\partial z_{nr}} & \frac{\partial h_{Rn}}{\partial z_{ni}} \\ \frac{\partial h_{In}}{\partial z_{1r}} & \frac{\partial h_{In}}{\partial z_{1i}} & \cdots & \frac{\partial h_{In}}{\partial z_{nr}} & \frac{\partial h_{In}}{\partial z_{ni}} \end{pmatrix}. \tag{19}$$

Without loss of generality, it is enough to analyze the stability of the point $\mathbf{z} = v\mathbf{e}_1 = (v, 0, \ldots, 0)$, which corresponds to $\mathbf{w} = v\mathbf{a}_1$. Now $v = v_r + iv_i$ and $|\mathbf{z}^H\mathbf{s}|^2 = |s_1|^2$. Evaluating the gradient (16) at point $\mathbf{z} = v\mathbf{e}_1$, we get

$$\nabla H(v\mathbf{e}_1) = 2 \begin{pmatrix} v_r E\{|s_1|^2\, g(|s_1|^2)\} \\ v_i E\{|s_1|^2\, g(|s_1|^2)\} \\ 0 \\ \vdots \\ 0 \end{pmatrix} \tag{20}$$

using the independence of s_j and the zero-mean and unit-variance properties of s_j.

For the Hessian at point $\mathbf{z} = v\mathbf{e}_1$ we use the independence of s_j and the assumptions $E\{\mathbf{s}\mathbf{s}^H\} = \mathbf{I}$ and $E\{\mathbf{s}\mathbf{s}^T\} = \mathbf{O}$, yielding

$$\nabla^2 H(v\mathbf{e}_1) = 2 \begin{pmatrix} E\{|s_1|^2 g(|s_1|^2) + 2v_r^2|s_1|^4 g'(|s_1|^2)\} & 2v_r v_i E\{|s_1|^4 g'(|s_1|^2)\} & 0 & \cdots & 0 \\ 2v_r v_i E\{|s_1|^4 g'(|s_1|^2)\} & E\{|s_1|^2 g(|s_1|^2) + 2v_i^2|s_1|^4 g'(|s_1|^2)\} & 0 & \cdots & 0 \\ 0 & 0 & \alpha & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & \alpha \end{pmatrix} \tag{21}$$

where

$$\alpha = E\{g(|s_1|^2) + |s_1|^2 g'(|s_1|^2)\}. \tag{22}$$

Note that we do *not* assume that the real and imaginary parts of the same variable s_j are independent, even though we use the independence of s_j and s_k, $j \neq k$ as discussed in Sec. 2.

Now we make a small perturbation $\boldsymbol{\varepsilon} = (\varepsilon_{1r}, \varepsilon_{1i}, \ldots, \varepsilon_{nr}, \varepsilon_{ni})$ where ε_{jr} and ε_{ji} are the real and imaginary parts of $\varepsilon_j \in \mathbb{C}$ and evaluate the Taylor expansion of H:

$$\begin{aligned} H(v\mathbf{e}_1 + \boldsymbol{\varepsilon}) &= H(v\mathbf{e}_1) + \boldsymbol{\varepsilon}^T \nabla H(v\mathbf{e}_1) + \frac{1}{2}\boldsymbol{\varepsilon}^T \nabla^2 H(v\mathbf{e}_1)\boldsymbol{\varepsilon} + o(\|\boldsymbol{\varepsilon}\|^2) \\ &= H(\mathbf{e}_1) + 2(\varepsilon_{1r}v_r + \varepsilon_{1i}v_i)E\{|s_1|^2 g(|s_1|^2)\} + \varepsilon_{1r}^2 E\{|s_1|^2 g(|s_1|) + 2v_r^2|s_1|^4 g'(|s_1|^2)\} \\ &\quad + 4v_r v_i \varepsilon_{1r}\varepsilon_{1i} E\{|s_1|^4 g'(|s_1|^2)\} + \varepsilon_{1i}^2 E\{|s_1|^2 g(|s_1|) + 2v_i^2|s_1|^4 g'(|s_1|^2)\} \\ &\quad + E\{g(|s_1|^2) + |s_1|^2 g'(|s_1|^2)\} \sum_{j>1} (\varepsilon_{jr}^2 + \varepsilon_{ji}^2) + o(\|\boldsymbol{\varepsilon}\|^2)\,. \end{aligned} \tag{23}$$

Furthermore, due to the constraint $\|\mathbf{w}\| = 1$ and thus $\|v\mathbf{e}_1 + \boldsymbol{\varepsilon}\| = 1$ we get

$$2(\varepsilon_{1r}v_r + \varepsilon_{1i}v_i) = -\sum_{j=1}^{n} (\varepsilon_{jr}^2 + \varepsilon_{ji}^2)\,. \tag{24}$$

Using this, we get

$$\begin{aligned} H(v\mathbf{e}_1 + \boldsymbol{\varepsilon}) &= H(v\mathbf{e}_1) + E\{g(|s_1|^2) + |s_1|^2 g'(|s_1|^2) \\ &\quad - |s_1|^2 g(|s_1|^2)\} \sum_{j>1} (\varepsilon_{jr}^2 + \varepsilon_{ji}^2) \\ &\quad + 2(\varepsilon_{1r}v_r + \varepsilon_{1i}v_i)^2 E\{|s_1|^4 g'(|s_1|^2)\} \\ &\quad + o(\|\boldsymbol{\varepsilon}\|^2) \end{aligned} \tag{25}$$

where the term of order $(\varepsilon_{1r}v_r + \varepsilon_{1i}v_i)^2$ is $o(\|\boldsymbol{\varepsilon}\|^2)$ according to (24), giving

$$\begin{aligned} H(v\mathbf{e}_1 + \boldsymbol{\varepsilon}) &= H(v\mathbf{e}_1) + E\{g(|s_1|^2) + |s_1|^2 g'(|s_1|^2) \\ &\quad - |s_1|^2 g(|s_1|^2)\} \sum_{j>1} (\varepsilon_{jr}^2 + \varepsilon_{ji}^2) \\ &\quad + o(\|\boldsymbol{\varepsilon}\|^2)\,. \end{aligned} \tag{26}$$

Thus $\mathbf{z} = v\mathbf{e}_1$ is an extremum, and it is the maximum (minimum) if

$$E\{g(|s_1|^2) + |s_1|^2 g'(|s_1|^2) - |s_1|^2 g(|s_1|^2)\} < 0 \quad (> 0, \text{ resp.})\,. \tag{27}$$

Appendix B

Derivation of the algorithm

We shall derive the fixed-point algorithm for one unit. Let $w = w_r + iw_i$ and $x = x_r + ix_i$. For the ease of derivations, the algorithm updates the real and imaginary parts of w separately. We assume that the source signals s_j are white, i.e., they are zero-mean and have unit variances and uncorrelated real and imaginary parts of equal variances, that is, $E\{\mathbf{s}\mathbf{s}^H\} = \mathbf{I}$ and $E\{\mathbf{s}\mathbf{s}^T\} = \mathbf{O}$. The observed variable $\mathbf{x}$ is whitened so that it also obeys $E\{\mathbf{x}\mathbf{x}^H\} = \mathbf{I}$.

According to the Kuhn-Tucker conditions, the optima of $E\{G(|\mathbf{w}^H\mathbf{x}|^2)\}$ under the constraint $E\{|\mathbf{w}^H\mathbf{x}|^2\} = \|\mathbf{w}\|^2 = 1$ are obtained at points where

$$\nabla E\{G(|\mathbf{w}^H\mathbf{x}|^2)\} - \beta \nabla E\{|\mathbf{w}^H\mathbf{x}|^2\} = 0 \tag{28}$$

where $\beta \in \mathbb{R}$ and the gradient is computed with respect to real and imaginary parts of w separately. The first term in (28) is

$$\begin{aligned} \nabla E\{G(|\mathbf{w}^H\mathbf{x}|^2)\} &= \begin{pmatrix} \dfrac{\partial}{\partial w_{1r}} \\ \dfrac{\partial}{\partial w_{1i}} \\ \vdots \\ \dfrac{\partial}{\partial w_{nr}} \\ \dfrac{\partial}{\partial w_{ni}} \end{pmatrix} E\{G(|\mathbf{w}^H\mathbf{x}|^2)\} \\ &= 2 \begin{pmatrix} E\{\mathrm{Re}\{x_1(\mathbf{w}^H\mathbf{x})^*\}\, g(|\mathbf{w}^H\mathbf{x}|^2)\} \\ E\{\mathrm{Im}\{x_1(\mathbf{w}^H\mathbf{x})^*\}\, g(|\mathbf{w}^H\mathbf{x}|^2)\} \\ \vdots \\ E\{\mathrm{Re}\{x_n(\mathbf{w}^H\mathbf{x})^*\}\, g(|\mathbf{w}^H\mathbf{x}|^2)\} \\ E\{\mathrm{Im}\{x_n(\mathbf{w}^H\mathbf{x})^*\}\, g(|\mathbf{w}^H\mathbf{x}|^2)\} \end{pmatrix} \end{aligned} \tag{29}$$

and the second term in (28) is

$$\nabla E\{|\mathbf{w}^H\mathbf{x}|^2\} = 2\begin{pmatrix} \mathrm{Re}\{w_1\} \\ \mathrm{Im}\{w_1\} \\ \vdots \\ \mathrm{Re}\{w_n\} \\ \mathrm{Im}\{w_n\} \end{pmatrix} \tag{30}$$

where the assumption $E\{\mathbf{x}\mathbf{x}^H\} = \mathbf{I}$ was used.

The Newton method is used to solve (28). The Jacobian matrix of $\nabla E\{G(|\mathbf{w}^H\mathbf{x}|^2)\}$ as in (29) can be approximated as

$$\begin{aligned} &\nabla^2 E\{G(|\mathbf{w}^H\mathbf{x}|^2)\} \\ &\quad = 2E\{(\nabla^2|\mathbf{w}^H\mathbf{x}|^2)g(|\mathbf{w}^H\mathbf{x}|^2) \\ &\qquad + 2(\nabla|\mathbf{w}^H\mathbf{x}|^2)(\nabla|\mathbf{w}^H\mathbf{x}|^2)^T g'(|\mathbf{w}^H\mathbf{x}|^2)\} \quad (31) \\ &\quad \approx 2E\{g(|\mathbf{w}^H\mathbf{x}|^2) + |\mathbf{w}^H\mathbf{x}|^2 g'(|\mathbf{w}^H\mathbf{x}|^2)\}\mathbf{I} \quad (32) \end{aligned}$$

where the approximation was done by separating the expectations. Also, $E\{\mathbf{x}\mathbf{x}^T\} = \mathbf{O}$ (which follows straightforward from $E\{\mathbf{s}\mathbf{s}^T\} = \mathbf{O}$) was used. The Jacobian matrix of $\beta\nabla E\{|\mathbf{w}^H\mathbf{x}|^2\}$ is, using (30),

$$\beta\nabla^2 E\{|\mathbf{w}^H\mathbf{x}|^2\} = 2\beta\mathbf{I}\,. \tag{33}$$

The total approximative Jacobian of (28) is now

$$\mathbf{J} = 2(E\{g(|\mathbf{w}^H\mathbf{x}|^2) + |\mathbf{w}^H\mathbf{x}|^2 g'(|\mathbf{w}^H\mathbf{x}|^2)\} - \beta)\mathbf{I} \tag{34}$$

which is diagonal and thus easy to invert. We obtain the following approximative Newton iteration:

$$\begin{aligned} \mathbf{w}^+ &= \mathbf{w} - \frac{E\{\mathbf{x}(\mathbf{w}^H\mathbf{x})^* g(|\mathbf{w}^H\mathbf{x}|^2)\} - \beta\mathbf{w}}{E\{g(|\mathbf{w}^H\mathbf{x}|^2) + |\mathbf{w}^H\mathbf{x}|^2 g'(|\mathbf{w}^H\mathbf{x}|^2)\} - \beta} \\ \mathbf{w}_{\mathrm{new}} &= \frac{\mathbf{w}^+}{\|\mathbf{w}^+\|}\,. \end{aligned} \tag{35}$$

If we multiply both sides of (35) by $\beta - E\{g(|\mathbf{w}^H\mathbf{x}|^2) + |\mathbf{w}^H\mathbf{x}|^2 g'(|\mathbf{w}^H\mathbf{x}|^2)\}$, the fixed-point algorithm simplifies to

$$\begin{aligned} \mathbf{w}^+ &= E\{\mathbf{x}(\mathbf{w}^H\mathbf{x})^* g(|\mathbf{w}^H\mathbf{x}|^2)\} \\ &\quad - E\{g(|\mathbf{w}^H\mathbf{x}|^2) + |\mathbf{w}^H\mathbf{x}|^2 g'(|\mathbf{w}^H\mathbf{x}|^2)\}\mathbf{w} \\ \mathbf{w}_{\mathrm{new}} &= \frac{\mathbf{w}^+}{\|\mathbf{w}^+\|}\,. \end{aligned} \tag{36}$$

Decorrelation schemes suitable for deflationary or symmetric separation of the independent components were presented in Sec. 5.

References

1. P. Comon 1994, "Independent component analysis — a new concept?" *Signal Processing* **36**, 287–314.
2. C. Jutten and J. Herault 1991, "Blind separation of sources, part I: An adaptive algorithm based on neuromimetic architecture," *Signal Processing* **24**, 1–10.
3. A. Hyvärinen 1999, "Fast and robust fixed-point algorithms for independent component analysis," *IEEE Transactions on Neural Networks* **10**(3), 626–634.
4. A. Hyvärinen and E. Oja 1997, "A fast fixed-point algorithm for independent component analysis," *Neural Computation* **9**, 1483–1492.
5. A. D. Back and A. C. Tsoi 1994, "Blind deconvolution of signals using a complex recurrent network," in *Neural Networks for Signal Processing 4, Proceedings of the 1994 IEEE Workshop*, eds. J. Vlontzos, J. Hwang and E. Wilson (IEEE Press), pp. 565–574.
6. E. Moreau and O. Macchi 1994, "Complex self-adaptive algorithms for source separation based on higher order contrasts," in *Proc. VII European Signal Processing Conference* (*EUSIPCO'94*), Vol. II, 1157–1160, Edinburgh, Scotland, September.
7. C. L. Nikias and A. P. Petropulu 1993, *Higher-Order Spectra Analysis. A Nonlinear Signal Processing Framework* (Prentice-Hall).
8. C. W. Therrien 1992, *Discrete Random Signals and Statistical Signal Processing* (Prentice-Hall).
9. A. Hyvärinen and E. Oja 1998, "Independent component analysis by general nonlinear Hebbian-like learning rules," *Signal Processing* **64**, 301–313.
10. A. Hyvärinen and P. Hoyer 2000, "Emergence of phase and shift invariant features by decomposition of natural images into independent feature subspaces," *Neural Computation* (in press).
11. J.-F. Cardoso 1998, "Multidimensional independent component analysis," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing* (*ICASSP'98*), Seattle, USA.

PUBLICATION 2

Ella Bingham and Heikki Mannila. Random projection in dimensionality reduction: applications to image and text data. In Foster Provost and Ramakrishnan Srikant, editors, *Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 245–250, San Francisco, CA, USA, August 2001.

Random projection in dimensionality reduction: Applications to image and text data

Ella Bingham and Heikki Mannila*
Laboratory of Computer and Information Science
Helsinki University of Technology
P.O. Box 5400, FIN-02015 HUT, Finland
ella@iki.fi, Heikki.Mannila@hut.fi

ABSTRACT

Random projections have recently emerged as a powerful method for dimensionality reduction. Theoretical results indicate that the method preserves distances quite nicely; however, empirical results are sparse. We present experimental results on using random projection as a dimensionality reduction tool in a number of cases, where the high dimensionality of the data would otherwise lead to burdensome computations. Our application areas are the processing of both noisy and noiseless images, and information retrieval in text documents. We show that projecting the data onto a random lower-dimensional subspace yields results comparable to conventional dimensionality reduction methods such as principal component analysis: the similarity of data vectors is preserved well under random projection. However, using random projections is computationally significantly less expensive than using, e.g., principal component analysis. We also show experimentally that using a sparse random matrix gives additional computational savings in random projection.

Keywords

random projection, dimensionality reduction, image data, text document data, high-dimensional data

1. INTRODUCTION

In many applications of data mining, the high dimensionality of the data restricts the choice of data processing methods. Such application areas include the analysis of market basket data, text documents, image data and so on; in these cases the dimensionality is large due to either a wealth of alternative products, a large vocabulary, or the use of large image windows, respectively. A statistically optimal way of dimensionality reduction is to project the data onto a lower-dimensional orthogonal subspace that captures as much of the variation of the data as possible. The best (in mean-square sense) and most widely used way to do this is principal component analysis (PCA); unfortunately it is quite expensive to compute for high-dimensional data sets. A computationally simple method of dimensionality reduction that does not introduce a significant distortion in the data set would thus be desirable.

In random projection (RP), the original high-dimensional data is projected onto a lower-dimensional subspace using a random matrix whose columns have unit lengths. RP has been found to be a computationally efficient, yet sufficiently accurate method for dimensionality reduction of high-dimensional data sets. While this method has attracted lots of interest, empirical results are sparse.

In this paper we give experimental results on using RP as a dimensionality reduction tool on high-dimensional image and text data sets. In both application areas, random projection is compared to well known dimensionality reduction methods. We show that despite the computational simplicity of random projection, it does not introduce a significant distortion in the data.

The data sets used in this paper are of very different natures. Our image data is from monochrome images of natural scences. An image is presented as a matrix of pixel brightness values, the distribution of which is generally approximately Gaussian: symmetric and bell-shaped. Text document data is presented in vector space [25], in which each document forms one d-dimensional vector where d is the vocabulary size. The i-th element of the vector indicates (some function of) the frequency of the i-th vocabulary term in the document. Document data is often highly sparse or peaked: only some terms from the vocabulary are present in one document, and most entries of the document vector are zero. Also, document data has a nonsymmetric, positively skewed distribution, as the term frequencies are nonnegative. It is instructive to see how random projection works as a dimensionality reduction tool in the context of these two very different application areas.

We also present results on images corrupted by noise, and our experimental results indicate that random projection is not sensitive to impulse noise. Thus random projection is a promising alternative to some existing methods in noise reduction (e.g. median filtering), too.

This paper is organized as follows. At the end of this introduction we discuss related work on random projections and similarity search. Section 2 presents different dimensionality

*On leave at Nokia Research Center

KDD 01 San Francisco CA USA

reduction methods. Section 3 gives the experimental results of dimensionality reduction on image data, and Section 4 on text data. Finally, Section 5 gives a conclusion.

1.1 Related work

Papadimitriou et al. [22] use random projection in the preprocessing of textual data, prior to applying LSI. They present experimental results on an artificially generated set of documents. In their approach, the columns of the random projection matrix are assumed strictly orthogonal, but actually this need not be the case, as we shall see in our experiments.

Kaski [17, 16] has presented experimental results in using the random mapping in the context of the WEBSOM[1] system. Kurimo [20] applies random projection to the indexing of audio documents, prior to using LSI and SOM. Kleinberg [19] and Indyk and Motwani [14] use random projections in nearest-neighbor search in a high dimensional Euclidean space, and also present theoretical insights. Dasgupta [6, 7] has used random projections in learning high-dimensional Gaussian mixture models. Other applications of random projection include e.g. [4, 28].

The problems of dimensionality reduction and similarity search have often been addressed in the information retrieval literature, and other approaches than random projection have been presented. Ostrovsky and Rabani [21] give a dimension reduction operation that is suitable for clustering. Agrawal et al. [3] map time series into frequency domain by the discrete Fourier transform and only retain the first few frequencies. Keogh and Pazzani [18] reduce the dimension of time series data by segmenting the time series into sections and indexing only the section means. Aggarwal et al. [2] index market basket data by a specific signature table, which easens the similarity search. Wavelet transforms ([12, 27] etc.) are a common method of signal compression.

2. METHODS FOR DIMENSIONALITY REDUCTION

2.1 Random projection

In random projection, the original d-dimensional data is projected to a k-dimensional ($k << d$) subspace through the origin, using a random $k \times d$ matrix R whose columns have unit lengths. Using matrix notation where $X_{d\times N}$ is the original set of N d-dimensional observations,

$$X^{RP}_{k\times N} = R_{k\times d} X_{d\times N} \tag{1}$$

is the projection of the data onto a lower k-dimensional subspace. The key idea of random mapping arises from the Johnson-Lindenstrauss lemma [15]: if points in a vector space are projected onto a randomly selected subspace of suitably high dimension, then the distances between the points are approximately preserved. For a simple proof of this result, see [10, 8].

Random projection is computationally very simple: forming the random matrix R and projecting the $d \times N$ data matrix X into k dimensions is of order $O(dkN)$, and if the data matrix X is sparse with about c nonzero entries per column, the complexity is of order $O(ckN)$ [22].

Strictly speaking, (1) is not a projection because R is generally not orthogonal. A linear mapping such as (1) can cause significant distortions in the data set if R is not orthogonal. Orthogonalizing R is unfortunately computationally expensive. Instead, we can rely on a result presented by Hecht-Nielsen [13]: in a high-dimensional space, there exists a much larger number of almost orthogonal than orthogonal directions. Thus vectors having random directions might be sufficiently close to orthogonal, and equivalently $R^T R$ would approximate an identity matrix. In our experiments, the mean squared difference between $R^T R$ and an identity matrix was about $1/k$ per element.

When comparing the performance of random projection to that of other methods of dimensionality reduction, it is instructive to see how the similarity of two vectors is distorted in the dimensionality reduction. We measure the similarity of data vectors either as their Euclidean distance or as their inner product. In the case of image data, Euclidean distance is a widely used measure of similarity. Text documents, on the other hand, are generally compared according to the cosine of the angle between the document vectors; if document vectors are normalized to unit length, this corresponds to the inner product of the document vectors.

We write the Euclidean distance between two data vectors x_1 and x_2 in the original large-dimensional space as $||x_1 - x_2||$. After the random projection, this distance is approximated by the scaled Euclidean distance of these vectors in the reduced space:

$$\sqrt{d/k}\,||Rx_1 - Rx_2|| \tag{2}$$

where d is the original and k the reduced dimensionality of the data set. The scaling term $\sqrt{d/k}$ takes into account the decrease in the dimensionality of the data: according to the Johnson-Lindenstrauss lemma, the expected norm of a projection of a unit vector onto a random subspace through the origin is $\sqrt{k/d}$ [15].

The choice of the random matrix R is one of the key points of interest. The elements r_{ij} of R are often Gaussian distributed, but this need not be the case. Achlioptas [1] has recently shown that the Gaussian distribution can be replaced by a much simpler distribution such as

$$r_{ij} = \sqrt{3} \cdot \begin{cases} +1 & \text{with probability } \frac{1}{6} \\ 0 & \text{with probability } \frac{2}{3} \\ -1 & \text{with probability } \frac{1}{6}. \end{cases} \tag{3}$$

In fact, practically all zero mean, unit variance distributions of r_{ij} would give a mapping that still satisfies the Johnson-Lindenstrauss lemma. Achlioptas' result means further computational savings in database applications, as the computations can be performed using integer arithmetics. In our experiments we shall use both Gaussian distributed random matrices and sparse matrices (3), and show that Achlioptas' theoretical result indeed has practical significance. In context of the experimental results, we shall refer to RP when the projection matrix is Gaussian distributed and SRP when the matrix is sparse and distributed according to (3). Otherwise, the shorthand RP refers to any random projection.

2.2 PCA, SVD and LSI

In principal component analysis (PCA), the eigenvalue decomposition of the data covariance matrix is computed as $\mathrm{E}\{XX^T\} = E\Lambda E^T$ where the columns of matrix E are the eigenvectors of the data covariance matrix $\mathrm{E}\{XX^T\}$ and Λ is a diagonal matrix containing the respective eigenvalues.

[1] See `http://websom.hut.fi/websom/`

If dimensionality reduction of the data set is desired, the data can be projected onto a subspace spanned by the most important eigenvectors:

$$X^{PCA} = E_k^T X \tag{4}$$

where the $d \times k$ matrix E_k contains the k eigenvectors corresponding to the k largest eigenvalues. PCA is an optimal way to project data in the mean-square sense: the squared error introduced in the projection is minimized over all projections onto a k-dimensional space. Unfortunately, the eigenvalue decomposition of the data covariance matrix (whose size is $d \times d$ for d-dimensional data) is very expensive to compute. The computational complexity of estimating the PCA is $O(d^2N) + O(d^3)$ [11]. There exists computationally less expensive methods [26, 24] for finding only a few eigenvectors and eigenvalues of a large matrix; in our experiments, we use appropriate Matlab routines to realize these.

A closely related method is singular value decomposition (SVD): $X = USV^T$ where orthogonal matrices U and V contain the left and right singular vectors of X, respectively, and the diagonal of S contains the singular values of X. Using SVD, the dimensionality of the data can be reduced by projecting the data onto the space spanned by the left singular vectors corresponding to the k largest singular values:

$$X^{SVD} = U_k^T X \tag{5}$$

where U_k is of size $d \times k$ and contains these k singular vectors. Like PCA, SVD is also expensive to compute. There exists numerical routines such as the power or the Lanczos method [5] that are more efficient than PCA for sparse data matrices X, and that is why we shall use SVD instead of PCA in the context of sparse text document data. For a sparse data matrix $X_{d \times N}$ with about c nonzero entries per column, the computational complexity of SVD is of order $O(dcN)$ [22].

Latent semantic indexing (LSI) [9, 22] is a dimensionality reduction method for text document data. Using LSI, the document data is presented in a lower-dimensional "topic" space: the documents are characterized by some underlying (latent, hidden) concepts referred to by the terms. LSI can be computed either by PCA or SVD of the data matrix of N d-dimensional document vectors.

2.3 Discrete cosine transform

Discrete cosine transform (DCT) is a widely used method for image compression and as such it can also be used in dimensionality reduction of image data. DCT is computationally less burdensome than PCA and its performance approaches that of PCA. DCT is also optimal for human eye: the distortions introduced occur at the highest frequencies only, and the human eye tends to neglect these as noise. DCT can be performed by simple matrix operations [23, 27]: an image is transformed to the DCT space and dimensionality reduction is done in the inverse transform by discarding the transform coefficients corresponding to the highest frequencies. Computing the DCT is not data-dependent, in contrast to PCA that needs the eigenvalue decomposition of data covariance matrix; that is why DCT is orders of magnitude cheaper to compute than PCA. Its computational complexity is of the order $O(dN \log_2(dN))$ for a data matrix of size $d \times N$ [27].

3. RESULTS ON IMAGE DATA

The data set consisted of $N = 1000$ image windows drawn from 13 monochrome images[2] of natural scenes.The sizes of the original images were 256×256 pixels, and windows of size 50×50 were randomly drawn from the images. Each image window was presented as one d-dimensional column vector ($d = 2500$).

3.1 Noiseless image data

When comparing different methods for dimensionality reduction, the criteria are the amount of distortion caused by the method and its computational complexity. In the case of image data we measure the distortion by comparing the Euclidean distance between two dimensionality reduced data vectors to their Euclidean distance in the original high-dimensional space. In the case of random projection, the Euclidean distance in the reduced space is scaled as shown in (2); with other methods, no scaling is performed.

We first tested the effect of the reduced dimensionality using different values of k in $[1, 800]$. At each k, the dimensionality reducing matrix operation was computed anew. Figure 1 shows the error in the distance between members of a pair of data vectors, averaged over 100 pairs. The results of random projection with a Gaussian distributed random matrix (RP), random projection with a sparse random matrix as in (3) (SRP), principal component analysis (PCA) and discrete cosine transform (DCT) are shown, together with their 95 per cent confidence intervals.

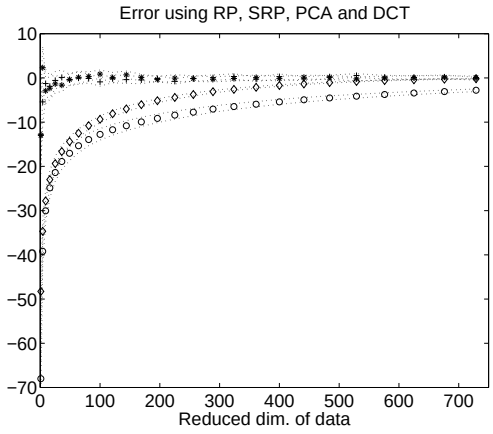

Figure 1: The error produced by RP (+), SRP (∗), PCA (⋄) and DCT (∘) on image data, and 95 % confidence intervals over 100 pairs of data vectors.

In Figure 1 it is clearly seen that random projection (RP and SRP) yields very accurate results: dimensionality reduction by random projection does not distort the data significantly more than PCA. At dimensions $k > 600$, random projection and PCA give quite accurate results but the error produced by DCT is clearly visible. At smaller dimensions also PCA distorts the data. This tells us that the variation in the data is mostly captured by the first 600 principal components, because the error in PCA is dependent on the sum of omitted eigenvalues, and k is equal to the number of eigen-

[2]Available from
`http://www.cis.hut.fi/projects/ica/data/images/`

values retained. In contrast, the random projection method continues to give accurate results until $k = 10$. One explanation for the success of random projection is the J-L scaling term $\sqrt{d/k}$ (Formula (2)), which takes into account the decrease in the dimensionality. In PCA, such scaling would only be useful in the smallest dimensions but a straightforward rule is difficult to give.

Another point of interest is the computational complexity of the methods. Figure 2 shows the number of Matlab's floating point operations needed when using RP, SRP, PCA or DCT in dimensionality reduction, in a logarithmic scale. It can be seen that PCA is significantly more burdensome than random projection or DCT. (In the case of DCT, only the chosen data vectors were transformed instead of the whole data set; this makes the number of floating point operations rather small.)

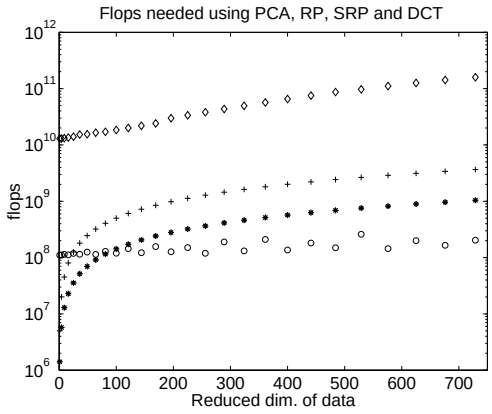

Figure 2: Number of Matlab's floating point operations needed when reducing the dimensionality of image data using RP (+), SRP (∗), PCA (⋄) and DCT (∘), in a logarithmic scale.

From Figures 1–2 we can conclude that random projection is a computationally inexpensive method of dimensionality reduction while preserving the distances of data vectors practically as well as PCA and clearly better than DCT. Even more, at smallest dimensions RP outperforms both PCA and DCT.

Dimensionality reduction on image data differs slightly from another common procedure, image compression, in which the image is transformed into a more economical form for e.g. transmission, and then transformed back into the original space. The transformation is often chosen so that the resulting image looks as similar as possible to the original image, to a human eye. In this respect, the discrete cosine transform has proven optimal. To see how an image whose dimensionality is reduced by RP would look like, the random mapping should be inverted. The pseudoinverse of R is expensive to compute, but since R is almost orthogonal, the transpose of R is a good approximation of the pseudoinverse, and the image can be computed as $X^{new}_{d\times N} = R^T_{d\times k} X^{RP}_{k\times N}$ where X^{RP} is the result of the random projection (1). Nonetheless, the obtained image is visually worse than a DCT compressed image, to a human eye. Thus random projection is successful in applications where the distance or similarity between data vectors should be preserved under dimensionality reduction as well as possible, but where the data is not intended to be visualized for the human eye. These applications include, e.g., machine vision: it would be possible to automatically detect whether an (on-line) image from a surveillance camera has changed or not.

3.2 Noise reduction in images

In our second set of experiments we considered noisy images. The images were corrupted by salt-and-pepper impulse noise: with probability 0.2, a pixel in the image was turned black or white. We wanted to project the data in such a way that the distance between two data vectors in the reduced noisy data space would be as close as possible to the distance between these vectors in the high-dimensional *noiseless* data space, even though the dimensionality reduction was applied to high-dimensional noisy images.

A simple yet effective way of noise reduction especially in the case of salt-and-pepper impulse noise is median filtering (MF) where each pixel in the image is replaced by the median of the pixel brightnesses in its neighborhood. The median is not affected by individual noise spikes and so median filtering eliminates impulse noise quite well [27]. A common neighborhood size is 3×3 pixels which was also used in our experiments. MF is computationally very efficient, of order $O(dmN)$ for N image windows of d pixels, where m denotes the size of the neighborhood (in our case, $m = 9$). Also, MF does not require dimensionality reduction; thus its result can be used as a yardstick when comparing methods for dimensionality reduction and noise cancellation.

Figure 3 shows how the distance between two noisy image windows is distorted in dimensionality reduction, compared to their distance in the original high-dimensional, noiseless space. Here we can compare different dimensionality reduction methods with respect to their sensitivity to noise. We can see that median filtering introduces quite a large distortion in the image windows, despite that to a human eye it removes impulse noise very efficiently. The distortion is due to blurring: pixels are replaced by the median of their neighborhood, eliminating noise but also small details. PCA, DCT and random projection perform quite similarly to the noiseless case. From Figure 3 we can conclude that random projection is a promising alternative to dimensionality reduction on noisy data, too, as it does not seem to be sensitive to impulse noise. There exists of course many other methods for noise reduction, too. Here our interest was mainly in dimensionality reduction and not noise reduction.

4. RESULTS ON TEXT DATA

Next, we applied dimensionality reduction techniques on text document data from four newsgroups of the 20 newsgroups corpus[3]: sci.crypt, sci.med, sci.space and soc.religion.christian. The documents were converted into term frequency vectors and some common terms were removed using McCallum's Rainbow toolkit[4] but no stemming was used.

The data was not made zero mean, nor was the overall variance of entries of the data matrix normalized. The document vectors were only normalized to unit length. This kind of preprocessing was different from that applied to im-

[3]Available from `http://www.cs.cmu.edu/~textlearning`
[4]Available from `http://www.cs.cmu.edu/~mccallum/bow`

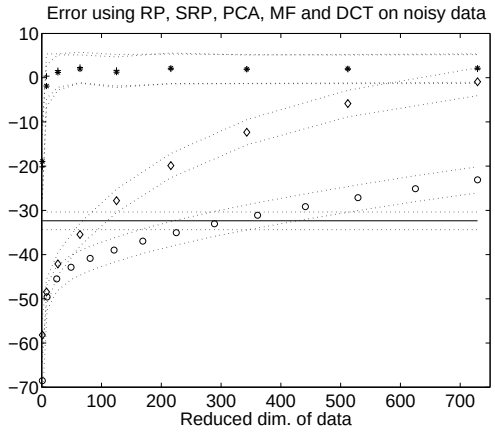

Figure 3: The error produced by RP (+), SRP (∗), PCA (⋄), DCT (○) and MF (−) on noisy image data, with 95% confidence intervals over 100 pairs of image windows. In MF dimensionality is not reduced.

age data. Together with the distinct natures of image and text data, differences in preprocessing yielded slightly different results on these different data sets. The size of the vocabulary was $d = 5000$ terms and the data set consisted of $N = 2262$ newsgroup documents.

We randomly chose pairs of data vectors (that is, documents) and computed their similarity as their inner product. The error in the dimensionality reduction was measured as the difference between the inner products before and after the dimensionality reduction.

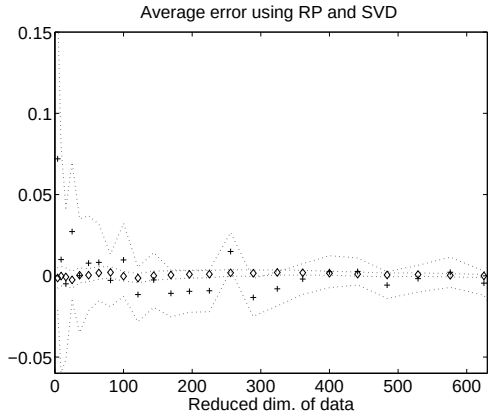

Figure 4: The error produced by RP (+) and SVD (⋄) on text document data, with 95% confidence intervals over 100 pairs of document vectors.

Figure 4 shows the error introduced by dimensionality reduction. The results are averaged over 100 document pairs. The results of SVD and random projection with a Gaussian distributed random matrix are shown, together with 95 per cent confidence intervals. The reduced dimensionality k took values in $[1, 700]$. It is seen that random projection is not quite as accurate as SVD but in many applications the error may be neglectable. The Johnson-Lindenstrauss result [15] states that Euclidean distances are retained well in random projection. The case of inner products is a different one — Euclidean distances of document vectors would probably have been preserved better. It is a common practice to measure the similarity of document vectors by their inner products; thus we present results on them.

Despite using efficient SVD routines for finding a few singular vectors of a sparse matrix, SVD is still orders of magnitude more burdensome than RP.

Our results on text document data indicate that random projection can be used in dimensionality reduction of large document collections, with less computational complexity than latent semantic indexing (SVD). Similarly to what was presented in [22], RP can speed up latent semantic indexing (LSI): the dimensionality of the data is first reduced by RP and the burdensome LSI is only computed in the new low-dimensional space. In [22] the documents were generated artificially and the random matrix R was assumed strictly orthogonal; our experiments show that neither of these restrictions is actually necessary. Another common problem in text document retrieval is query matching. Random projection might be useful in query matching if the query is long, or if a set of *similar* documents instead of one particular document were searched for.

5. CONCLUSIONS

We have presented new and promising experimental results on random projection in dimensionality reduction of high-dimensional real-world data sets. When comparing different methods for dimensionality reduction, the criteria are the amount of distortion caused by the method and its computational complexity. Our results indicate that random projection preserves the similarities of the data vectors well even when the data is projected to moderate numbers of dimensions; the projection is yet fast to compute.

Our application areas were of quite different natures: noisy and noiseless images of natural scenes, and text documents from a newsgroup corpus. In both application areas, random projection proved to be a computationally simple method of dimensionality reduction, while still preserving the similarities of data vectors to a high degree.

We also presented experimental results of random projection using a sparsely populated random matrix introduced in [1]. It is in fact not necessary to use a Gaussian distributed random matrix but much simpler matrices still obey the Johnson-Lindenstrauss lemma [15], giving computational savings.

One should emphasize that random projection is beneficial in applications where the distances of the original high-dimensional data points are meaningful as such — if the original distances or similarities are themselves suspect, there is little reason to preserve them. For example, consider using the data in neural network training. Projecting the data onto a lower dimensional subspace speeds up the training only if the training is based on interpoint distances; such problems include clustering and k Nearest Neighbors etc. Also, consider the significance of each of the dimensions of a data set. In a Euclidean space, every dimension is equally important and independent of the others, whereas e.g. in a process monitoring application some measured quantities (that is, dimensions) might be closely related to others, and the interpoint distances do not necessarily bear a clear meaning.

A still more realistic application of random projection would be to use it in a data mining problem, e.g. clustering, and compare the results and computational complexity of mining the original high-dimensional data and dimensionality reduced data; this is a topic of a further study.

An interesting open problem concerns k, the number of dimensions needed for random projections. The Johnson-Lindenstrauss result [15, 10, 8] gives bounds that are much higher than the ones that suffice to give good results on our empirical data. For example, in the case of our image data, the lower bound for k on $\epsilon = 0.2$ is 1600 but in the experiments, $k \approx 50$ was enough. The Johnson-Lindenstrauss result, of course, is a worst-case one, and it would be interesting to understand which properties of our experimental data make it possible to get good results by using fewer dimensions.

We conclude that random projection is a good alternative to traditional, statistically optimal methods of dimensionality reduction that are computationally infeasible for high dimensional data. Random projection does not suffer from the curse of dimensionality, quite contrary to the traditional methods.

6. REFERENCES

[1] D. Achlioptas. Database-friendly random projections. In *Proc. ACM Symp. on the Principles of Database Systems*, pages 274–281, 2001.

[2] C. C. Aggarwal, J. L. Wolf, and P. S. Yu. A new method for similarity indexing of market basket data. In *Proc. 1999 ACM SIGMOD Int. Conf. on Management of data*, pages 407–418, 1999.

[3] R. Agrawal, C. Faloutsos, and A. Swami. Efficient similarity search in sequence databases. In *Proc. 4th Int. Conf. of Data Organization and Algorithms*, pages 69–84. Springer, 1993.

[4] R. I. Arriaga and S. Vempala. An algorithmic theory of learning: robust concepts and random projection. In *Proc. 40th Annual Symp. on Foundations of Computer Science*, pages 616–623. IEEE Computer Society Press, 1999.

[5] M.-W. Berry. Large-scale sparse singular value computations. *International Journal of Super-Computer Applications*, 6(1):13–49, 1992.

[6] S. Dasgupta. Learning mixtures of Gaussians. In *40th Annual IEEE Symp. on Foundations of Computer Science*, pages 634–644, 1999.

[7] S. Dasgupta. Experiments with random projection. In *Proc. Uncertainty in Artificial Intelligence*, 2000.

[8] S. Dasgupta and A. Gupta. An elementary proof of the Johnson-Lindenstrauss lemma. Technical Report TR-99-006, International Computer Science Institute, Berkeley, California, USA, 1999.

[9] S. Deerwester, S.T. Dumais, G.W. Furnas, and T.K. Landauer. Indexing by latent semantic analysis. *Journal of the Am. Soc. for Information Science*, 41(6):391–407, 1990.

[10] P. Frankl and H. Maehara. The Johnson-Lindenstrauss lemma and the sphericity of some graphs. *Journal of Combinatorial Theory, Ser. B*, 44:355–362, 1988.

[11] G.H. Golub and C.F. van Loan. *Matrix Computations*. North Oxford Academic, Oxford, UK, 1983.

[12] A. Graps. An introduction to wavelets. *IEEE Computational Science and Engineering*, 2(2):50–61, 1995.

[13] R. Hecht-Nielsen. Context vectors: general purpose approximate meaning representations self-organized from raw data. In J.M. Zurada, R.J. Marks II, and C.J. Robinson, editors, *Computational Intelligence: Imitating Life*, pages 43–56. IEEE Press, 1994.

[14] P. Indyk and R. Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proc. 30th Symp. on Theory of Computing*, pages 604–613. ACM, 1998.

[15] W.B. Johnson and J. Lindenstrauss. Extensions of Lipshitz mapping into Hilbert space. In *Conference in modern analysis and probability*, volume 26 of *Contemporary Mathematics*, pages 189–206. Amer. Math. Soc., 1984.

[16] S. Kaski. Data exploration using self-organizing maps. In *Acta Polytechnica Scandinavica, Mathematics, Computing and Management in Engineering Series*, number 82. 1997. Dr.Tech. thesis, Helsinki University of Technology, Finland.

[17] S. Kaski. Dimensionality reduction by random mapping. In *Proc. Int. Joint Conf. on Neural Networks*, volume 1, pages 413–418, 1998.

[18] E. J. Keogh and M. J. Pazzani. A simple dimensionality reduction technique for fast similarity search in large time series databases. In *4th Pacific-Asia Conf. on Knowledge Discovery and Data Mining*, 2000.

[19] J.M. Kleinberg. Two algorithms for nearest-neighbor search in high dimensions. In *Proc. 29th ACM Symp. on Theory of Computing*, pages 599–608, 1997.

[20] M. Kurimo. Indexing audio documents by using latent semantic analysis and SOM. In E. Oja and S. Kaski, editors, *Kohonen Maps*, pages 363–374. Elsevier, 1999.

[21] R. Ostrovsky and Y. Rabani. Polynomial time approximation schemens for geometric k-clustering. In *Proc. 41st Symp. on Foundations of Computer Science*, pages 349–358. IEEE, 2000.

[22] C.H. Papadimitriou, P. Raghavan, H. Tamaki, and S. Vempala. Latent semantic indexing: A probabilistic analysis. In *Proc. 17th ACM Symp. on the Principles of Database Systems*, pages 159–168, 1998.

[23] K.R. Rao and P. Yip. *Discrete Cosine Transform: Algorithms, Advantages, Applications*. Academic Press, 1990.

[24] S. Roweis. EM algorithms for PCA and SPCA. In *Neural Information Processing Systems 10*, pages 626–632, 1997.

[25] G. Salton and M.J. McGill. *Introduction to modern information retrieval*. McGraw-Hill, 1983.

[26] L. Sirovich and R. Everson. Management and analysis of large scientific datasets. *Int. Journal of Supercomputer Applications*, 6(1):50–68, spring 1992.

[27] M. Sonka, V. Hlavac, and R. Boyle. *Image processing, analysis, and machine vision*. PWS Publishing, 1998.

[28] S. Vempala. Random projection: a new approach to VLSI layout. In *Proc. 39th Annual Symp. on Foundations of Computer Science*. IEEE Computer Society Press, 1998.

PUBLICATION 3

Aapo Hyvärinen and Ella Bingham. Connection between multilayer perceptrons and regression using independent component analysis. *Neurocomputing*, 50(C):211–222, January 2003.

ELSEVIER

Neurocomputing 50 (2003) 211–222

NEUROCOMPUTING

www.elsevier.com/locate/neucom

Connection between multilayer perceptrons and regression using independent component analysis

Aapo Hyvärinen*, Ella Bingham

Neural Networks Research Centre, Helsinki University of Technology, P.O. Box 5400, 02150 Espoo, Finland

Received 31 August 2000; accepted 23 November 2001

Abstract

The data model of independent component analysis (ICA) gives a multivariate probability density that describes many kinds of sensory data better than classical models like Gaussian densities or Gaussian mixtures. When only a subset of the random variables is observed, ICA can be used for regression, i.e. to predict the missing observations. In this paper, we show that the resulting regression is closely related to regression by a multi-layer perceptron (MLP). In fact, if linear dependencies are first removed from the data, regression by ICA is, as a first-order approximation, equivalent to regression by MLP. This theoretical result gives a new interpretation of the elements of the MLP: The outputs of the hidden layer neurons are related to estimates of the values of the independent components, and the sigmoid nonlinearities are obtained from the probability densities of the independent components.

Keywords: Nonlinear regression; Multilayer perception; Independent component analysis; Projection pursuit

1. Introduction

Independent component analysis (ICA) [2,11,6,13] is a recently developed statistical model where we express observed random variables $x_1, x_2, \ldots, x_q$ as linear combinations of unknown component variables, denoted by $s_1, s_2, \ldots, s_n$. The components s_i are, by definition, mutually statistically independent, and zero-mean. Let us arrange the observed variables x_i into a vector $\mathbf{x} = (x_1, x_2, \ldots, x_q)^{\mathrm{T}}$ and the independent components

* Corresponding author. Tel.: +358-9-451-3278; fax: +358-9-451-3277.
E-mail address: aapo.hyvarinen@hut.fi (A. Hyvärinen).
URL: http://www.cis.hut.fi/aapo/

0925-2312/03/$ - see front matter
PII: S0925-2312(01)00705-6

s_i into a vector $\mathbf{s}$, respectively; then the linear relationship is given by

$$\mathbf{x} = \mathbf{A}\mathbf{s}. \tag{1}$$

Here, $\mathbf{A}$ is an unknown $q \times n$ matrix, called the mixing matrix. The basic problem of ICA estimation is then to estimate the mixing matrix $\mathbf{A}$, as well as the densities of the s_i, using only observations of the mixtures x_j. This means that we try to approximate the joint density of $\mathbf{x}$ as precisely as possible by the densities of sums of independent random variables. We assume here that $n \geqslant q$, in order to have a nonsingular joint density.

Regression, i.e. prediction, is one of the fundamental problems in supervised learning. In the general regression problem, the variables in $\mathbf{x}$ are divided into two parts, observed and missing, that is, the predicting variables and the variables to be predicted. For simplicity, we can arrange the variables in $\mathbf{x}$ so that the k first variables form the vector of the observed variables $\mathbf{x}_\mathrm{o} = (x_1, \ldots, x_k)^\mathrm{T}$, and the remaining variables form the vector of the missing variables $\mathbf{x}_\mathrm{m} = (x_{k+1}, \ldots, x_q)^\mathrm{T}$. Thus the model can be written as

$$\begin{pmatrix} \mathbf{x}_\mathrm{o} \\ \mathbf{x}_\mathrm{m} \end{pmatrix} = \begin{pmatrix} \mathbf{A}_\mathrm{o} \\ \mathbf{A}_\mathrm{m} \end{pmatrix} \mathbf{s}. \tag{2}$$

The problem is now to predict $\mathbf{x}_\mathrm{m}$ for a given observation of $\mathbf{x}_\mathrm{o}$. To be able to predict the $\mathbf{x}_\mathrm{m}$, we must use (an estimate of) the joint probability distribution of $\mathbf{x}$. Of course, we must have some previous observations of $\mathbf{x}_\mathrm{m}$ to be able to estimate the joint probability distribution, that is, to be able to measure how the predicted (missing) variables depend on the predicting (observed) variables. (This is the case for any regression method.) The regression $\hat{\mathbf{x}}_\mathrm{m}$ is conventionally defined as the conditional expectation:

$$\hat{\mathbf{x}}_\mathrm{m} = E\{\mathbf{x}_\mathrm{m} | \mathbf{x}_\mathrm{o}\}. \tag{3}$$

Since the data model of ICA describes well some aspects of many kinds of sensory data [15], it would be natural to attempt to use ICA for regression for such data sets. In fact, since the ICA data model gives (an approximation of) the joint probability density of $\mathbf{x}$, it is straightforward, at least in principle, to first model the joint density of $\mathbf{x}$ by ICA, and then, for a given sample of incomplete data, predict the missing values in $\mathbf{x}_\mathrm{m}$ using the conditional expectation, which is well defined once the ICA model has been estimated. Thus, we obtain

$$E\{\mathbf{x}_\mathrm{m} | \mathbf{x}_\mathrm{o}\} = \mathbf{A}_\mathrm{m} \int_{\mathbf{A}_\mathrm{o}\mathbf{s}=\mathbf{x}_\mathrm{o}} \mathbf{s}\, p(\mathbf{s})\, \mathrm{d}\mathbf{s}. \tag{4}$$

In the following, we shall call this generic idea "regression by ICA".

Regression by ICA was already used in [14] to predict missing pixels in images. In [5], the method was considered in a more general setting, and it was proposed that instead of the conditional expectation, i.e. the minimum mean-square error estimator, one could use the maximum a posteriori estimator, which is computationally much simpler. A similar method was considered in [16], though the connection to ICA was not mentioned.

Regression by ICA is parametric,[1] yet nonlinear. It is, in fact, a direct generalization of ordinary linear regression: if the independent components s_i were Gaussian, Eq. (1) would simply give multivariate Gaussian distributions, and the conditional expectation would be a linear function of $\mathbf{x}_o$. Regression by ICA is also closely connected to projection pursuit regression [4], because it concentrates on those projections that are the most non-Gaussian. It could therefore be expected to partially avoid the curse of dimensions.

Thus, ICA gives us one approach to nonlinear regression. A vast literature on regression exists, however, both in neural network and statistics literature, and it would be most useful to know what is the connection between this regression by ICA and classical regression methods. The purpose of this paper is to show that an intimate connection exists between regression by ICA, and regression by multi-layer perceptrons whose structure closely mimics the structure of the ICA model. A two-layer MLP which has the same number of hidden units as the ICA model, and whose nonlinearity is equal to the so-called score function of the independent components gives, as a first-order approximation, the same regression as ICA. It is assumed here that linear dependencies are removed as a preprocessing step. This result gives a new interpretation of MLPs. Moreover, it shows clearly some further relations between regression by ICA and other regression methods.

Some preliminary results were reported in [7].

2. Regression by ICA and by an MLP: the connection

Before announcing our main result, we must discuss the preprocessing of the data. We assume here that the data is first linearly preprocessed so that any linearly predictable part of $\mathbf{x}_m$ is removed. In other words, the $\mathbf{x}_m$ are replaced by the residuals of linear regression. The result of this preprocessing step is that the $\mathbf{x}_o$ and $\mathbf{x}_m$ are uncorrelated. Second, the vectors $\mathbf{x}_o$ and $\mathbf{x}_m$ are each separately whitened. Note that these preprocessing steps cannot be replaced using ordinary whitening methods used in ICA, because they confound the division to observed (predicting) and missing (predicted) variables. As is usual in ICA, this particular form of whitening implies that $\mathbf{A}$ is an orthogonal matrix.

Our result is based on first-order approximations whose accuracy depends on the validity of some assumptions. First, the independent components must have distributions that are not too far from the Gaussian distribution; this critical assumption is discussed in Sections 4 and 5. Second, we assume that the dimension of $\mathbf{x}_o$ is large when compared to the dimension of $\mathbf{x}_m$; this assumption seems to be true in most practical cases where multivariate regression is applied.

Let us denote the probability densities of the s_i by p_i, and by $g_i(u) = p_i'(u)/p_i(u) + cu$ a function that equals the negative score function p_i'/p_i of the probability density of s_i,

[1] We assume here that the distributions of the independent components are either known or modelled by a density family of a limited number of parameters. In general, if the distributions of the independent components are not known, the regression would be semiparametric, though arguably weakly so.

plus an arbitrary linear term, which is the same for all i. For example, the tanh function is the score function of a mildly super-Gaussian (sparse) distribution [1]. Denote further by g the multi-dimensional function that consists of applying g_i on the ith component of its argument, for every i. After the above preprocessing and assumptions we have the following result (proven in Appendix A):

$$\boxed{E\{\mathbf{x}_m|\mathbf{x}_o\} \approx \mathbf{A}_m g(\mathbf{A}_o^T \mathbf{x}_o).} \tag{5}$$

In other words, the regression function for data modeled by ICA, is given by the output of an MLP with one hidden layer. The weight vectors of the MLP are simple functions of the mixing matrix, and the nonlinear activation functions of the MLP are functions of the probability densities of the s_i.

To get insight into this approximation, let us consider super-Gaussian densities, in which case we can take $g_i(u) = -\tanh(u) + u$ for all i. This is a shrinkage function [8] that approximately reduces the value of its argument by a given constant, resembling a soft-thresholding operation. Now, the vector $\mathbf{A}_o^T \mathbf{x}_o$ can be interpreted as an initial linear estimate of $\mathbf{s}$. (In fact, due to whitening, $\mathbf{A}$ is orthogonal and therefore $\mathbf{A}_o^T$ is equal to the pseudoinverse of $\mathbf{A}_o$.) Thus, the nonlinear aspect of (5) consists largely of *thresholding* the linear estimates of $\mathbf{s}$, to obtain $\hat{\mathbf{s}} = g(\mathbf{A}_o^T \mathbf{x}_o)$. The thresholding can be considered as a way of improving the linear estimate, in a manner similar to the denoising method in [8]. The final linear layer is basically a linear reconstruction of the form $\mathbf{x}_m = \mathbf{A}_m \hat{\mathbf{s}}$.

3. Relation to other methods

3.1. Projection pursuit regression

Our results make as well the connection of regression by ICA to projection pursuit regression quite explicit. Assume that the dimension of the data is very high, and that only certain projections of the data have non-Gaussian distributions. One variation of projection pursuit regression [4] would then consist of finding the most non-Gaussian projections, and using only those projections to construct the regression function. This can be intuitively justified as follows. Since all linear dependencies were removed as a preprocessing step, and the optimal regression for Gaussian data is linear, Gaussian projections of the data cannot give any new information that would be useful for regression, and thus it is sensible to concentrate on the non-Gaussian projections.

In fact, if we assume that some of the independent components are Gaussian (say, the last ones with indices $i = l+1, \ldots, r$), the regression function in (5) has the form

$$E\{\mathbf{x}_m|\mathbf{x}_o\} \approx \sum_{i=1}^{l} \mathbf{v}_i g_i(\mathbf{w}_i^T \mathbf{x}_o), \tag{6}$$

where $\mathbf{w}_i$ is the ith column of the matrix $\mathbf{A}_o$, and $\mathbf{v}_i$ is the ith column of the matrix $\mathbf{A}_m$. In this sum, only the l first linear estimates $\mathbf{w}_i^T \mathbf{x}_o$ of the independent components are used, i.e. only those corresponding to the non-Gaussian components. This is because

the linear score function of the Gaussian independent components can be taken equal to zero because of the possibility of adding an arbitrary linear term to the nonlinearities g_i. On the other hand, it is a well-known fact in the theory of ICA estimation that the projections in the most non-Gaussian directions give estimates of the independent components [6,11]. (This is not exactly true here, though, because we estimate the independent components using a smaller number of observed variables.) Thus, we see that the regression given in (5) is closely related to projection pursuit regression, both consisting of using component-wise nonlinearities in the most non-Gaussian directions.

3.2. Wavelet shrinkage

Regression by ICA is also closely related to wavelet shrinkage [3]. In wavelet shrinkage, the data is first transformed into the wavelet domain. In the regression context, any missing data points are treated as zeros. A thresholding operator is then applied on the wavelet coefficients, and the data is transformed back into the original domain. Consider, for example, prediction (reconstruction) of missing pixels in image data. The utility of such a reconstruction scheme can be intuitively seen in the following way: The linear reconstructions of wavelet coefficients are linear estimates of edges or bars; thresholding them makes edges and bars sharper in the reconstructed image.

It has been shown that the independent components of image windows are quite similar to the wavelet coefficients; the wavelet transform can be thus considered as an approximation of ICA [15,8]. As discussed above, the nonlinearity in the hidden layer of the MLP can be taken to be a thresholding function when the independent components are super-Gaussian, as usual with image data. Moreover, since the ICA transform is orthogonal due to whitening, the linear estimation of the independent components, as performed in the first layer of the MLP is equivalent to estimating the independent components as if the missing pixels were zero. Thus, we see that the regression by ICA, according to the approximation in (5), is very closely related to wavelet shrinkage for certain kinds of data, consisting of the same steps of transforming to sparse or independent components, thresholding, and inversion of the transform.

4. Simulations

We performed simple simulations to validate the accuracy of the approximations involved in our result. We generated artificially data according to the ICA model, and compared the true ICA regression with our approximation.

Our simulation data was 100-dimensional and there were $N = 101\,000$ data samples. The independent components, generated according to some probability density (see below) were mixed using a randomly generated $n \times n$ mixing matrix. The mixtures $\mathbf{x}$ were then divided into observed ($\mathbf{x}_o$) and missing ($\mathbf{x}_m$). The dimensionality of $\mathbf{x}_o$ was 99 and the dimensionality of $\mathbf{x}_m$ was 1. The latter was chosen to facilitate analysis and visualization of results.

In the preprocessing phase, the value of the missing variable $\mathbf{x}_m$ was first predicted by linear regression, and the residual of this regression was used in place of $\mathbf{x}_m$ in

the sequel. After this linear prediction, the variables in $\mathbf{x}_o$ were uncorrelated and their variance was set to one; similarly, the variance of $\mathbf{x}_m$ was set to one. Thus the data were whitened.

After the above preprocessing the data was divided in two sets, a training data set of size 100 000 and a test data set of size 1000. The ICA estimation on the training data set gave the estimated values for the source signals $\mathbf{s}$ and the mixing matrix $\mathbf{A} = \begin{pmatrix} \mathbf{A}_o \\ \mathbf{A}_m \end{pmatrix}$.

The test data set was used to compute estimates for the missing variable $\mathbf{x}_m$. The value of the missing variable $\mathbf{x}_m$ was predicted either using numerical integration as in (4), or using our approximation in (5). The success of the approximation was measured by the correlation coefficient between the two values. Furthermore, we computed the correlation coefficients between the true values of $\mathbf{x}_m$ are the results of numerical integration to see if the very principle of ICA regression is useful.

Three different distributions for the independent components were used, and the results were accumulated over 10 different random seeds.

In the following results, x_m denotes the true value of the missing variable, x_m^{num} is the estimated value computed by numerical integration, and x_m^{appr} is the value given by our MLP-like approximation

4.1. Strongly super-Gaussian data

In the first experiments the independent components $\mathbf{s}$ were generated according to the following strongly super-Gaussian density [8]:

$$p(s) = \frac{1}{2d} \frac{(\alpha+2)[\alpha(\alpha+1)/2]^{\alpha/2+1}}{[\sqrt{\alpha(\alpha+1)/2 + |s/d|}]^{\alpha+3}}, \tag{7}$$

where parameter values $\alpha = 1$ and $d = 1$ were chosen, giving

$$p(s) = \frac{1}{2} \frac{3}{(1+|s|)^4}. \tag{8}$$

The strong super-Gaussianity of this distribution is seen in the fact that the kurtosis is infinite. The score function of this probability density is

$$f'(s) = \frac{(\alpha+3)/d \operatorname{sign}(s)}{\sqrt{\alpha(\alpha+1)/2 + |s/d|}}. \tag{9}$$

The correlation coefficient between the numerical integration result and our approximation $\rho(x_m^{num}, x_m^{appr})$ was equal to 0.9067, which shows that the approximation was quite good. The scatterplot is shown in Fig. 1a. Interestingly, if we used the—tanh nonlinearity instead of the true score function (not shown), the correlation coefficient increased to 0.9303, probably because this is numerically more stable, avoiding the singularity at 0.

As for the success of the very principle of predicting the actual values of x_m, the correlation coefficient between the true x_m and the numerical integration $\rho(x_m, x_m^{num})$ was 0.9044, which shows that the very principle of ICA regression was feasible: using the ICA model in the regression does indeed give a good regression. This seems to be due to the strong super-Gaussianity of the s_i. The scatterplot is shown in Fig. 1b.

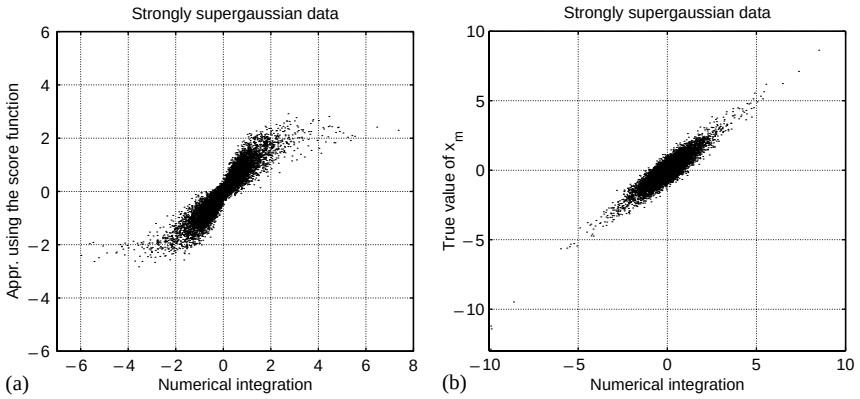

Fig. 1. The results for strongly super-Gaussian data: (a) scatterplot of optimal regression by numerical integration vs. regression using our approximation and (b) scatterplot of optimal regression by numerical integration vs. true values of x_m.

4.2. Laplace distributed data

In the second set of experiments the **s** were generated according to the Laplace distribution:

$$p(s) = \frac{\exp(-\sqrt{2}|s|)}{\sqrt{2}} \tag{10}$$

for which the score function is

$$f'(s) = \sqrt{2}\,\mathrm{sign}(s). \tag{11}$$

The correlation coefficient between the numerical integration result and our approximation $\rho(x_m^{num}, x_m^{appr})$ was equal to 0.9120, which shows that the approximation was quite good (see Fig. 2a).

On the other hand, the estimator x_m^{num} obtained by numerical integration correlates rather poorly with the true value of the missing variable x_m: the correlation coefficient is only 0.6489 (see Fig. 2b). Thus, ICA regression does not work that well in this case. This is probably because its success depends on the non-Gaussianity of the s_i, and thus requires the s_i to be strongly non-Gaussian. Likewise, the MLP-like approximation is not very successful in predicting the true value of the missing variable, the correlation coefficient being 0.5843.

4.3. Very weakly super-Gaussian data

In the third set of experiments the latent variables **s** were generated according to the Cosh distribution:

$$p(s) = \frac{1}{2}\,\frac{1}{\cosh^2 s} \tag{12}$$

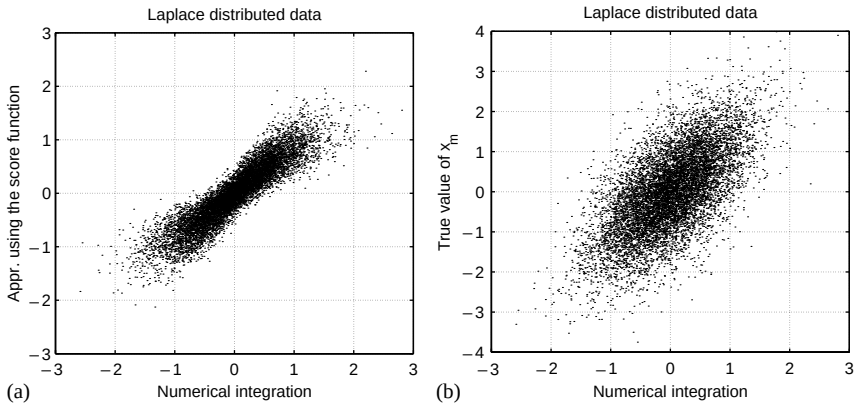

Fig. 2. The results for Laplace (moderately super-Gaussian) data: (a) scatterplot of optimal regression by numerical integration vs. regression using our approximation and (b) scatterplot of optimal regression by numerical integration vs. true values of x_m.

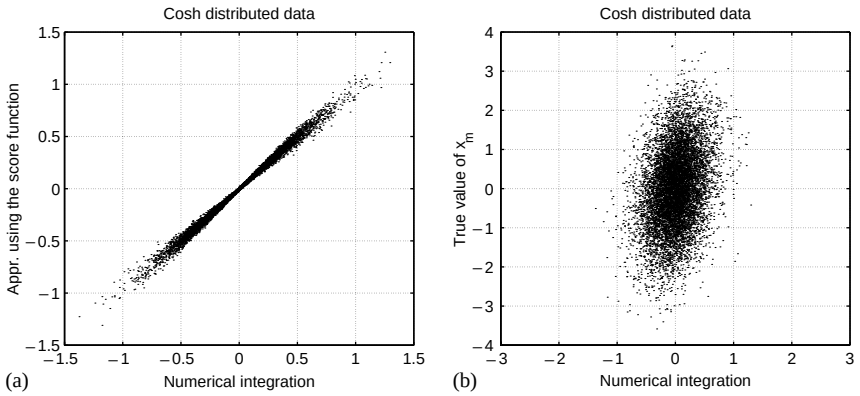

Fig. 3. The results for weakly super-Gaussian data: (a) scatterplot of optimal regression by numerical integration vs. regression using our approximation and (b) scatterplot of optimal regression by numerical integration vs. true values of x_m.

for which the score function is

$$f'(s) = \tanh s. \tag{13}$$

With this weakly super-Gaussian data, our approximation of the regression function was very good, the correlation coefficient being 0.9965. This was in fact to be expected: Our approximation was a first-order approximation in the vicinity of the Gaussian distribution for the s_i, and therefore it is not surprising that it works best when the s_i have almost Gaussian distributions. The scatterplot is in Fig. 3a.

On the other hand, we see again that the principle of ICA regression itself does not work well at all due to the weak non-Gaussianity of the data. The correlation

coefficient between the optimal regression computed by numerical integration and the true values of x_m was only 0.2969 (see Fig. 3b). Therefore, the approximating MLP cannot really predict the x_m, either, the correlation coefficient was 0.2954.

4.4. Conclusion

Thus, we see that our approximation works reasonably well. If the distributions of the independent components are close to Gaussian, it gives excellent results. If they are strongly super-Gaussian, the approximation is less accurate but still quite reasonable in the range we experimented with.

Another point is whether ICA regression in itself gives good regression results. Here we consider the prediction of the residuals of linear regression, since linear regression is a standard procedure and does not require the use of non-Gaussian structure. If the data simply does not contain enough structure, even the optimal regression method fails. We saw that the stronger the super-Gaussianity, the better the quality of the regression. For strongly super-Gaussian components, the values can be predicted quite well. In contrast, for weakly super-Gaussian components, ICA regression does not really explain the data; this is natural since for Gaussian data any regression beyond the linear one is impossible.

5. Discussion

We have shown a close connection between regression by ICA and regression by MLPs. Instead of developing a new method either for ICA estimation or nonlinear regression, our main contribution clearly lies in the theoretical insight on what multi-layer perceptrons are doing.

We showed that the output of each hidden-layer neuron in an MLP corresponds to the estimate of one independent component. This means that the problem of choosing the number of hidden units is somewhat equivalent to choosing the number of independent components in the ICA model. Thus, this classical problem in MLP research can be seen as a problem of choosing the model order, which is a classical problem in statistical modeling. Likewise, the choice of the nonlinearity is seen to be basically a problem of estimating the probability densities of the independent components.[2] Further, overlearning in MLPs can be seen to correspond to modeling the data with too many independent components, which is a form of overlearning typical of ICA [12]. To avoid overlearning, regularization is often used in MLPs, and similarly, regularizing the mixing matrix in ICA could be most useful [10].

Regression by ICA is, in practice, computationally demanding, due to the (possibly multi-dimensional) integration in (4). Our theoretical result might thus have some

[2] Note that the nonlinearities given by the score functions need not be known a priori: they can be estimated, just like the mixing matrix, by methods developed in ICA research, see [9]. The same is true for the number of independent components, though this is a much more difficult problem and satisfactory solutions may not be available [9].

practical significance, since it shows that the integration may be approximated by the computationally simple procedure of computing the outputs of an MLP.

It must be noted, however, that the equivalence we have shown is only true as a first-order approximation, for weakly non-Gaussian independent components. Only experiments can show whether this approximation is good enough in a given real-life application. Our simulations indicate that the approximation might quite well be useful. A second, independent question is, whether the very principle of ICA regression is useful in practice. Again, our simulations indicate that this might be so, if the independent components are strongly non-Gaussian, but assessing the utility in a real-life situation needs real-life experiments. In fact, we have a kind of contradiction: the approximation is based on the assumption that the components are weakly non-Gaussian, but the concept of regression by ICA seems to work only if the components are strongly non-Gaussian. However, the simulations above seem to indicate that our approximation is not bad even for strongly non-Gaussian variables. The assumption of weak non-Gaussianity could thus be considered as a technical assumption, allowing the derivation of an approximation that seems to be valid even for the more relevant case of strongly non-Gaussian components.

In conclusion, our result shows that the regression performed by MLPs, which is conventionally considered as nonparametric or semiparametric, can be interpreted in the framework of ICA as a model-based regression.

Appendix A. Proof of (5)

Denote $h_i(s_i) = s_i^2/2 - \frac{1}{2}\log 2\pi + \log p_i(s_i)$. The variances of the s_i are equal to one by definition. Due to the assumption of near-Gaussianity, $h_i(s_i)$ can thus be considered infinitesimal. We can write

$$E\{\mathbf{x}_m|\mathbf{x}_o\} = \mathbf{A}_m \frac{1}{(2\pi)^{n/2}} \int_{\mathbf{x}_o=\mathbf{A}_o\mathbf{s}} \mathbf{s} \exp\left(\sum_i [-s_i^2/2 + h_i(s_i)]\right) d\mathbf{s}. \tag{A.1}$$

Now, let us do a first-order approximation of $\sum_i h_i(s_i)$ in the vicinity of the point $\mathbf{A}_o^T\mathbf{x}_o$, i.e. the linear estimate of the independent components. This point is a linear approximation of the point where $p(\mathbf{s}|\mathbf{x}_o)$ is maximized. These approximations are likely to be rather exact if the dimension of $\mathbf{x}_o$ is large and the dimension of $\mathbf{x}_m$ is small. We obtain

$$E\{\mathbf{x}_m|\mathbf{x}_o\} \approx \mathbf{A}_m \frac{1}{(2\pi)^{n/2}} \cdot \int_{\mathbf{x}_o=\mathbf{A}_o\mathbf{s}} \mathbf{s} \exp\left(\sum_i [-s_i^2/2 + H_i(\mathbf{w}_i^T\mathbf{x}_o) + h_i'(\mathbf{w}_i^T\mathbf{x}_o)(s_i - \mathbf{w}_i^T\mathbf{x}_o)]\right) d\mathbf{s}, \tag{A.2}$$

where $\mathbf{w}_i$ denotes the ith column of $\mathbf{A}_o$. Now we can use the fact that $\exp(h_i(\mathbf{w}_i^T\mathbf{x}_o))$ is of order $1 + O(h)$. We can ignore this constant, since any change it could make

would be infinitesimal. Further, let us denote the constant $\exp(\sum_i -h'(\mathbf{w}_i^T\mathbf{x}_o)\mathbf{w}_i^T\mathbf{x}_o)$ by c_1. Thus we have

$$E\{\mathbf{x}_m|\mathbf{x}_o\} \approx \mathbf{A}_m \frac{c_1}{(2\pi)^{n/2}} \int_{\mathbf{x}_o=\mathbf{A}_o\mathbf{s}} \mathbf{s}\exp(-\|\mathbf{s}\|^2/2 + h'(\mathbf{A}_o^T\mathbf{x}_o)^T\mathbf{s}])\,d\mathbf{s}$$

$$\approx \mathbf{A}_m \frac{c_1c_2}{(2\pi)^{n/2}} \int_{\mathbf{x}_o=\mathbf{A}_o\mathbf{s}} \mathbf{s}\exp\left(-\frac{1}{2}\|\mathbf{s} - h'(\mathbf{A}_o^T\mathbf{x}_o)\|^2\right) d\mathbf{s}, \tag{A.3}$$

where h denotes the function where the h_i are applied componentwise. Here we have defined the constant $c_2 = \exp(\sum_i h_i'(\mathbf{w}_i^T\mathbf{x}_o)^2)$.

Thus, we have only a Gaussian integral left. It can be evaluated by making a norm-preserving variable change that parameterizes the space of the $\mathbf{s}$ such that $\mathbf{x}_o = \mathbf{A}_o\mathbf{s}$. This is given as $\mathbf{A}_o^T\mathbf{x}_o + \mathbf{A}_m^T\mathbf{u}$ where $\mathbf{u}$ is not constrained. Thus we obtain

$$\frac{c_1c_2}{(2\pi)^{n/2}} \int_{\mathbf{x}_o=\mathbf{A}_o\mathbf{s}} \mathbf{s}\exp\left(-\frac{1}{2}\|\mathbf{s} - h'(\mathbf{A}_o^T\mathbf{x}_o)\|^2\right) d\mathbf{s}$$

$$= \frac{c_1c_2}{(2\pi)^{n/2}} \int_{\mathbf{u}} [\mathbf{A}_o^T\mathbf{x}_o + \mathbf{A}_m^T\mathbf{u}]\exp\left(-\frac{1}{2}\|[\mathbf{A}_o^T\mathbf{x}_o + \mathbf{A}_m^T\mathbf{u}] - h'(\mathbf{A}_o^T\mathbf{x}_o)\|^2\right) d\mathbf{u}$$

$$= \frac{c_1c_2}{(2\pi)^{n/2}} \int_{\mathbf{u}} [\mathbf{A}_o^T\mathbf{x}_o + \mathbf{A}_m^T\mathbf{u}]\exp\left(-\frac{1}{2}[\|\mathbf{x}_o\|^2 + \|\mathbf{u}\|^2 + \|h'(\mathbf{A}_o^T\mathbf{x})\|^2 - 2h'(\mathbf{A}_o^T\mathbf{x}_o)^T\mathbf{A}_o^T\mathbf{x}_o - 2h'(\mathbf{A}_o^T\mathbf{x}_o)^T\mathbf{A}_m^T\mathbf{u}]\right) d\mathbf{u}$$

$$= \frac{c_1c_2}{(2\pi)^{n/2}} \exp(-\|\mathbf{x}_o\|^2/2 + h'(\mathbf{A}_o^T\mathbf{x}_o)^T\mathbf{A}_o^T\mathbf{x}_o) \cdot \int_{\mathbf{u}} [\mathbf{A}_o^T\mathbf{x}_o + \mathbf{A}_m^T\mathbf{u}]\exp\left(-\frac{1}{2}\|\mathbf{u} - \mathbf{A}_m h'(\mathbf{A}_o^T\mathbf{x}_o)\|^2\right) d\mathbf{u}, \tag{A.4}$$

where we have used the fact that the preprocessing implies that $\mathbf{A}_m\mathbf{A}_o^T = 0$ and $\mathbf{A}_m\mathbf{A}_m^T = \mathbf{A}_o\mathbf{A}_o^T = \mathbf{I}$. This can be evaluated by considering the Gaussian integral as an expectation of a Gaussian random vector. Furthermore, note that c_1 cancels the latter term in the exponential that is before the integral sign. Somewhat less rigorously, we could also assume that c_2 is approximately cancelled by the first term in that exponential; in any case this is only a scalar scaling. Thus, we obtain

$$\frac{c_1c_2}{(2\pi)^{n/2}} \int_{\mathbf{x}_o=\mathbf{A}_o\mathbf{s}} \mathbf{s}\exp\left(-\frac{1}{2}\|\mathbf{s} - h'(\mathbf{A}_o^T\mathbf{x}_o)\|^2\right) d\mathbf{s} \approx \mathbf{A}_o^T\mathbf{x}_o + \mathbf{A}_m^T\mathbf{A}_m h'(\mathbf{A}_o^T\mathbf{x}_o) \tag{A.5}$$

and we finally have

$$E\{\mathbf{x}_m|\mathbf{x}_o\} \approx \mathbf{A}_m h'(\mathbf{A}_o^T\mathbf{x}_o), \tag{A.6}$$

where we have again used the fact that the preprocessing implies that $\mathbf{A}_m\mathbf{A}_o^T = 0$. Here, $h_i'(u)$ is defined as $h_i'(u) = u + (\log p_i)'(u)$. On the other hand, $\mathbf{A}_m\mathbf{A}_o^T = 0$ implies that addition of any linear function to h' does not change the regression. Therefore, one

can take $h_i'(u) = (\log p_i)'(u) + cu$, i.e. h_i' can be defined as the negative score function of s_i plus any linear function. The linear function must be the same for all i.

References

[1] A.J. Bell, T.J. Sejnowski, An information-maximization approach to blind separation and blind deconvolution, Neural Comput. 7 (1995) 1129–1159.
[2] P. Comon, Independent component analysis—a new concept? Signal Process. 36 (1994) 287–314.
[3] D.L. Donoho, I.M. Johnstone, G. Kerkyacharian, D. Picard, Wavelet shrinkage: asymptopia? J. Roy. Statist. Soc. Ser. B 57 (1995) 301–337.
[4] P.J. Huber, Projection pursuit, Ann. Statist. 13 (2) (1985) 435–475.
[5] A. Hyvärinen, Sparse regression: utilizing the higher-order structure of data for prediction, in: Proceedings of the International Conference on Artificial Neural Networks (ICANN'98), Skövde, Sweden, 1998, pp. 541–546.
[6] A. Hyvärinen, Fast and robust fixed-point algorithms for independent component analysis, IEEE Trans. Neural Networks 10 (3) (1999) 626–634.
[7] A. Hyvärinen, Regression using independent component analysis, and its connection to multi-layer perceptrons, in: Proceedings of the International Conference on Artificial Neural Networks, Edinburgh, UK, 1999, pp. 491–496.
[8] A. Hyvärinen, Sparse code shrinkage: denoising of nongaussian data by maximum likelihood estimation, Neural Comput. 11 (7) (1999) 1739–1768.
[9] A. Hyvärinen, J. Karhunen, E. Oja, Independent Component Analysis, Wiley Interscience, New York, 2001.
[10] A. Hyvärinen, R. Karthikesh, Sparse priors on the mixing matrix in independent component analysis, in: Proceedings of the International Workshop on Independent Component Analysis and Blind Signal Separation (ICA2000), Helsinki, Finland, 2000, pp. 452–477.
[11] A. Hyvärinen, E. Oja, Independent component analysis: algorithms and applications, Neural Networks 13 (4–5) (2000) 411–430.
[12] A. Hyvärinen, J. Särelä, R. Vigário, Spikes and bumps: artefacts generated by independent component analysis with insufficient sample size, in: Proceedings of the International Workshop on Independent Component Analysis and Signal Separation (ICA'99), Aussois, France, 1999, pp. 425–429.
[13] C. Jutten, J. Hérault, Blind separation of sources, part I: An adaptive algorithm based on neuromimetic architecture, Signal Process. 24 (1991) 1–10.
[14] M. Lewicki, B. Olshausen, A probabilistic framework for the adaptation and comparison of image codes, J. Opt. Soc. Am. A: Opt. Image Sci. Vision 16 (7) (1998) 1587–1601.
[15] B.A. Olshausen, D.J. Field, Emergence of simple-cell receptive field properties by learning a sparse code for natural images, Nature 381 (1996) 607–609.
[16] Z. Roth, Y. Baram, Multidimensional density shaping by sigmoids, IEEE Trans. Neural Networks 7 (5) (1996) 1291–1298.

PUBLICATION 4

Ella Bingham, Ata Kabán, and Mark Girolami. Topic identification in dynamical text by complexity pursuit. *Neural Processing Letters*, 17(1):69–83, 2003.

Neural Processing Letters **17:** 69–83, 2003.

Topic Identification in Dynamical Text by Complexity Pursuit

ELLA BINGHAM★, ATA KABÁN† and MARK GIROLAMI‡
Neural Networks Research Centre, Helsinki University of Technology, P.O. Box 5400, FIN-02015 HUT, Finland

Abstract. The problem of analysing dynamically evolving textual data has arisen within the last few years. An example of such data is the discussion appearing in Internet chat lines. In this Letter a recently introduced source separation method, termed as *complexity pursuit*, is applied to the problem of finding topics in dynamical text and is compared against several blind separation algorithms for the problem considered. Complexity pursuit is a generalisation of projection pursuit to time series and it is able to use both higher-order statistical measures and temporal dependency information in separating the topics. Experimental results on chat line and newsgroup data demonstrate that the minimum complexity time series indeed do correspond to meaningful topics inherent in the dynamical text data, and also suggest the applicability of the method to query-based retrieval from a temporally changing text stream.

Key words. chat line discussion, complexity pursuit, dynamical text, independent component analysis, time series

Abbreviations. ICA – Independent component analysis; LSI – Latent semantic indexing

1. Introduction

In times of huge information flow especially in the Internet, there is a strong need for automatic textual data analysis tools. There are a number of algorithms and methods developed for text mining from static text collections [2]. The WEBSOM[1] is a document clustering and visualisation method [19]; its probabilistic counterpart has been presented e.g. in [16]. Another basic algorithm is Latent Semantic Indexing (LSI) [7] in which the data is projected onto a subspace spanned by the most important singular vectors of the data matrix; its probabilistic counterparts have been presented by Hofmann [9] and Papadimitriou [27]. LSI is found to capture some of the underlying semantics of textual data, resolving problems of synonymy and polysemy.

In recent years, the use of higher-order statistics and information-theoretic measures has gained popularity in the data analysis community. LSI uses only

★ Corresponding author. e-mail: ella@iki.fi
† Current address of correspondance: Department of Information Systems, Eötvös Loránd University, Pázmány Péter sétány 1/C, Budapest, Hungary H-1117. e-mail: kaba-ci0@paisley.ac.uk
‡ Current address of correspondance: School of Information and Communications Technology, University of Paisley, Paisley PA1 2BE, Scotland, UK. e-mail: giro-ci0@paisley.ac.uk
[1] See http://websom.hut.fi/websom/

second-order moments of the data and neglects any higher order correlations, so a natural step forward is to apply more powerful methods. An important class of higher order statistical methods are independent component analysis (ICA)-type methods [6, 12, 14]. In ICA a set of multidimensional observations is presented as a (linear) combination of some underlying latent features that are more or less independent of each other.

First approaches of using ICA in the context of text data were presented by Isbell and Viola [13], Kolenda et al. [22] and Kabán and Girolami [15]. In these approaches, the textual data is not a dynamic time series but rather an instantaneous mixture of independent topics. The underlying assumption which we also adopt is that the textual data consists of some more or less independent topics. In the text retrieval parlance, a *topic* is a probability distribution on the universe of terms; it is typically concentrated on terms that might be used when discussing a particular subject. In this paper, the word 'topic' also refers to a hidden, more or less independent random variable with time structure. Thus we can analyze the 'independent components' of text both by the terms they concentrate on, and by their activity in time.

Recently the issue of analyzing *dynamically evolving* textual data has arisen, and investigating appropriate tools for this task is of practical importance. An example of a dynamically evolving discussion is found in the Internet relay chat rooms. In these chat rooms daily news topics are discussed and the topic of interest changes according to participants' contributions. The online text stream can thus be seen as a time series, and methods of time series processing may be used to extract the underlying characteristics – here the topics – of the discussion. Kolenda and Hansen [20, 21] employ Molgedey and Schuster's [23] ICA algorithm for the identification of the dynamically evolving topics. Molgedey and Schuster's algorithm is an early separation algorithm which uses temporal information and does not require any higher order moments for the source separation problem. Kabán and Girolami [17] have recently presented a Hidden Markov Model (HMM)-type algorithm for the topographic visualization of time-varying data.

In this Letter a recently introduced powerful separating method is applied to the problem of extracting the topics of a dynamically evolving discussion. The method presented by Hyvärinen, termed as complexity pursuit [11], is a generalization of projection pursuit [8] to time series and it is able to exploit both higher-order and temporal dependency information in separating the topics. Complexity pursuit is a method for finding interesting projections of time series, the interestingness being measured as a *short coding length* of the projection. Projection pursuit, on the other hand, neglects any time-dependency information and defines interestingness as non-gaussianity. Complexity pursuit uses both information-theoretic measures and time-correlations of the data, which makes it more powerful and motivates its use in the task approached in this paper.

This paper is organized as follows. Section 2 describes the data and its preprocessing. Section 3 provides an introduction to complexity pursuit. Section 4 presents experimental results on using the complexity pursuit algorithm on chat line and

newsgroup data, and shows comparisons between several algorithms that have been presented for separating time-correlated signals. Finally, some conclusions are drawn in Section 5.

2. Dynamical Textual Data: Chat Line Discussion

Often the characteristics of the textual data of interest change over time. Such dynamical data can be found e.g. in the online news services. Our example of dynamically evolving text is chat line data, and later also newsgroup data that shares some similarities to chat line data.

The discussion found in chat lines on the Internet is an ongoing stream of text generated by the chat participants and the chat line moderator. To analyze it using data mining methods a convenient technique is to split the stream into windows that may be overlapping if desired. Each such window can now be viewed as one document. (In splitting the text stream, the boundaries between comment lines are not taken into account, as this might result into windows of different lengths. Also, this kind of partitioning is not always possible in other dynamical text streams, and we do not wish to restrict our analysis to chat line discussions only.)

We employ the vector space model [28] for representing the documents, although other models can be considered. In the vector space model, each document forms one T-dimensional vector where T is the number of distinct terms in the vocabulary. The i-th element of the vector indicates (some function of) the frequency of the i-th vocabulary term in the document. The data matrix $\mathbf{X}$, also called the term by document matrix, contains the document vectors as its columns and is of size $T \times N$ where N is the number of documents. We will write $\mathbf{X}$ when referring to the whole set of data vectors and $\mathbf{x}$ when referring to one of them; thus $\mathbf{X} = (\mathbf{x}(t))$, $t = 1, \ldots, N$.

As a preprocessing step we compute the LSI of the data matrix $\mathbf{X}$, that is, the singular value decomposition (SVD)

$$\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}^T \tag{1}$$

where orthogonal matrices $\mathbf{U}$ and $\mathbf{V}$ contain the left and right singular vectors of $\mathbf{X}$, respectively, and the pseudodiagonal matrix $\mathbf{D}$ contains the singular values of $\mathbf{X}$. The term by document matrix – which may be of very high dimension – is then projected onto a smaller dimensional subspace spanned by K left singular vectors in $\mathbf{U}_K$ corresponding to the K ($K \ll T$) largest singular values in the diagonal matrix $\mathbf{D}_K$:

$$\mathbf{Z} = \mathbf{D}_K^{-1}\mathbf{U}_K^T\mathbf{X}_K = \mathbf{V}_K^T \tag{2}$$

where $\mathbf{X}_K = \mathbf{U}_K\mathbf{D}_K\mathbf{V}_K^T$ is an approximation of $\mathbf{X}$. Thus the observations in $\mathbf{X}$ are represented as linear combinations of some orthogonal latent features. The new data matrix $\mathbf{Z} = \mathbf{V}_K^T$ and its columns $\mathbf{z}(t)$, $t = 1, \ldots, N$ are now the inputs for the algorithm that will be described in Section 3.

The time-structure of the topics of the discussion, or the minimum complexity projections, can be found by projecting $\mathbf{Z}$ onto the directions $\mathbf{W} = (\mathbf{w}_1 \cdots \mathbf{w}_M)$ given

by the complexity pursuit algorithm described in the following section. It is often advantageous to compute the LSI projection onto a somewhat larger dimensionality $K > M$ and then to find M minimum complexity projections.

To represent the estimated topics in the term space, the transpose of the original data is first projected onto the LSI term space by

$$\mathbf{Z}_{\text{term}} = \mathbf{D}_K^{-1}\mathbf{V}_K\mathbf{X}_K^T = \mathbf{U}_K^T \tag{3}$$

and then projected onto the directions $\mathbf{W}$ that were found earlier by feeding $\mathbf{Z}$ into the algorithm.

The LSI (SVD) preprocessing is computationally the most demanding part of the problem, of order $O(NTc)$ for a sparse $T \times N$ data matrix with c nonzero entries per column (here, c is the number of vocabulary terms present in one document). If new data is obtained after the LSI has been computed, the decomposition can be easily updated by folding-in [4] documents or terms: the LSI projection of a new document vector $\mathbf{x}_{\text{new}}$ (a new column in $\mathbf{X}$) is $\mathbf{z}_{\text{new}} = \mathbf{x}_{\text{new}}\mathbf{U}_K\mathbf{D}_K^{-1}$. Similarly, the projection of a new term vector $\mathbf{x}_{\text{new}}^{\text{term}}$ (a new row in $\mathbf{X}$) is $\mathbf{z}_{\text{new}}^{\text{term}} = \mathbf{x}_{\text{new}}^{\text{term}}\mathbf{V}_K\mathbf{D}_K^{-1}$.

3. The Complexity Pursuit Algorithm

Complexity pursuit [11] is a recently developed, computationally simple algorithm for separating interesting components from time series. It is an extension of projection pursuit [8] to time series data and also closely related to ICA. Projection pursuit seeks for directions in which the data has an interesting, structured distribution, the interestingness being understood as nongaussianity – neglecting any time-dependency information that may exist in the data. ICA, on the other hand, finds statistically independent directions. It is to be noted that under some restrictions, it is also possible to estimate the independent components using the time dependency information alone (see e.g. [3, 23]); however the early algorithms as that proposed in [23] do not utilize the distribution of the data in obtaining the separation. A heuristic way of combining both of these estimation criteria (nongaussianity and time-correlations) has been proposed in the JADE$_{TD}$ algorithm [24]. However, complexity pursuit combines these criteria in a principled way by employing the information theoretical concept of Kolmogoroff complexity [25] and developing a simple approximation of it. In complexity pursuit the structure of the projected time series is measured as the coding complexity. Time series which have the lowest coding complexity are considered the most interesting. Another method of separating independent sources in time series has recently been presented by Stone [30]; in his approach, it is assumed that the source signals are more predictable than any linear mixture of them. In Section 4 we shall present experimental results on using complexity pursuit, JADE$_{TD}$, ordinary ICA and the methods presented in [30] and [20]. Some other methods for detecting the semantics in a dynamical text stream are described e.g. in [29].

Our data model assumes that the observations $\mathbf{x}(t)$ are linear mixtures of some latent components:

$$\mathbf{x} = \mathbf{A}\mathbf{s} \tag{4}$$

where $\mathbf{x} = (x_1, \ldots, x_T)$ is the vector of observed random variables, $\mathbf{s} = (s_1, \ldots, s_M)$ is the vector of independently predictable latent components, and $\mathbf{A}$ is an unknown constant mixing matrix. In the context of complexity pursuit we do not put any special emphasis on the statistical independence of s_i, even though the data model (4) is similar to that of linear ICA.

A separate autoregressive model is assumed to model each component $s_i = \mathbf{w}_i^T\mathbf{x}$; as a simple special case of the algorithm presented in [11], we employ a first order autoregressive (AR) process $\hat{s}_i(t) = \alpha_i s_i(t - \tau)$ on each latent variable s_i. The approximate Kolmogoroff complexity of the residuals $\delta s(t) = s(t) - \hat{s}(t)$ (using the predictive coding of the components) [11]

$$\hat{K}(\delta(\mathbf{w}^T\mathbf{x}(t))) = E\Big\{G\Big(\frac{1}{\sigma_\delta(\mathbf{w})}\mathbf{w}^T(\mathbf{x}(t) - \alpha\mathbf{x}(t - \tau))\Big)\Big\} + \log\sigma_\delta(\mathbf{w}) \tag{5}$$

is then minimized, where G is the negative log-density of the residuals. In the above formula it is emphasized that the values of α and the residual standard deviation σ_δ depend on the projection vector $\mathbf{w}$ only. An additional constraint $E\{(\mathbf{w}^T\mathbf{x}(t))^2\} = 1$ is also required to fix the scale of the projection. In the right hand side of Formula (5) the first term measures the contribution of the nongaussianity, and the second term the contribution of the variance to the entropy of the residual. Minimizing the first term would find the direction of maximal nongaussianity of the residual, and minimizing the second term the direction of maximum autocovariances, i.e. maximum time-dependencies [11].

In our application the latent time-components s_i will model the evolving topics of the discussion. To find the minima of (5), the data is first whitened by LSI as described in the previous section. We denote by $\mathbf{z}(t)$ this preprocessed data, and $\mathbf{w}$ now corresponds to an estimate of a row of the inverse of the mixing matrix for whitened data. At every step of the algorithm, the autoregressive constant $\alpha(\mathbf{w})$ for the time series given by $\mathbf{w}^T\mathbf{z}(t)$ is first found using [11]

$$\hat{\alpha} = \mathbf{w}^T E\{\mathbf{z}(t)\mathbf{z}(t - \tau)\}\mathbf{w} \tag{6}$$

Then the gradient update of $\mathbf{w}$ that minimizes (5) is the following [11]:

$$\mathbf{w} \leftarrow \mathbf{w} - \mu E\{(\mathbf{z}(t) - \alpha(\mathbf{w})\mathbf{z}(t - \tau))g(\mathbf{w}^T(\mathbf{z}(t) - \alpha(\mathbf{w})\mathbf{z}(t - \tau)))\} \tag{7}$$

$$\mathbf{w} \leftarrow \mathbf{w}/||\mathbf{w}|| \tag{8}$$

The function g is chosen according to the probability distribution of the residual: to be exact, g should be the negative score function p'/p of the density of the residual, as g is the derivative of G in (5). In practice, the choice of g is quite flexible. Choosing a linear g corresponds to neglecting the higher-order structure of the data, and

analyzing the time-correlations of the signals only. This kind of complexity minimization is discussed e.g. in [26]. In general, a nonlinear g should be preferred for the estimation of nongaussian latent variables or residuals.

To estimate several projections one can either use a deflation scheme, or estimate all projections simultaneously in a symmetric manner and use orthogonal decorrelation

$$\mathbf{W} \leftarrow \sqrt{(\mathbf{W}\mathbf{W}^T)^{-1}}\mathbf{W} \tag{9}$$

instead of (8). In the deflationary approach, after the estimation of p projections, we run the algorithm for $\mathbf{w}_{p+1}$ and after every iteration step subtract from $\mathbf{w}_{p+1}$ the projections of the previously estimated p vectors, and then renormalize $\mathbf{w}_{p+1}$. This kind of Gram-Schmidt decorrelation is presented e.g. in [10].

The algorithm scales as $O(NK^2M)$ on preprocessed data; this is linear in the number of observations N as typically $K \ll N$ and $M \leqslant K$.

4. Experimental Results

4.1. EXPERIMENTAL SETTING

The chat line data used in our experiments was collected from the CNN Newsroom chat line[2]. A contiguous stream of almost 24 hr of discussion of 3200 chat participants, contributing 25 000 comment lines, was recorded on January 18th, 2001. The data was cleaned by omitting all user names and non-user generated text. The remaining text stream was split into windows of 12 rows (about 130 words); subsequent windows shared an overlap of 66%. From these windows a term histogram was generated using the Bow toolkit[3]. Stemming, stop-word removal and tf-idf (term frequency – inverse document frequency) term weighting were part of the process. This resulted in a term by document matrix $\mathbf{X}$ of size $T \times N = 5000 \times 7430$.

The binary valued coding of the term by document matrix – ith entry of a document vector was 1 if the ith vocabulary term was present in the document, and 0 otherwise – was used in the experiments. Binary coding avoids serious outliers in the data and is computationally simple; also, it may be suitable for short documents where the size of the vocabulary is large, such as short windows of chat line discussion.

The text document data is typically very sparse; in our chat line data, on the average, each document had about 40 vocabulary terms and only 0.65% of the entries of the data matrix $\mathbf{X}$ were nonzero. Sparsity gives additional computational savings, so we did not make the data zero mean as is often done in the context of ICA-type

[2]http://www.cnn.com/chat/channel/cnn_newsroom
[3]http://www.cs.cmu.edu/~mccallum/bow/

algorithms – that would have destroyed the sparsity and resulted in severe computational difficulties in the LSI preprocessing stage.

The choice of the number of estimated topics M is somewhat arbitrary[4]. It has been proved in [27] that if the data has a clear clustered structure, it is enough to choose M equal to the number of clusters. In our application the case is somewhat more complex, because more than one topic may be discussed at any one time, and real-life data may not have clear clusters.

The identified topics lend themselves easily to human evaluation if they are presented in the term space as described in the end of Section 2 and the most representative words associated with each $\mathbf{w}_i$, $i = 1, \ldots, M$ are listed. In the case of static data – e.g. ICA of functional magnetic resonance imaging (fMRI) and image recognition, or textual document analysis [15] – one can use both $\mathbf{X}$ and $\mathbf{X}^T$ for training (see [15] for derivation); this is called spatio-temporal ICA. In our case, the documents evolve dynamically but the terms have no time structure, and thus they will be employed in the visualization phase only.

It should also be noted that the projections $\mathbf{w}^T\mathbf{z}(t)$ that represent the latent topics of discussion are found by the complexity pursuit algorithm up to permutation, sign and scaling, as is always the case in the context of ICA-type algorithms. Therefore some prior knowledge based post-processing is necessary for interpreting the results. We know that the terms belonging to each topic should have a positively skewed distribution – there are often only a few terms that occur very frequently and correspondingly a large number of seldom occurring terms. (Katz [18] studies the distribution of words in phrases in more detail.) We must change the sign of the negatively skewed projections $\mathbf{w}^T\mathbf{z}(t)$ so that their distribution becomes positively skewed.

Our experiments showed that choosing a first order AR model $\hat{s}(t) = \alpha s(t - \tau)$ was successful and that lags of e.g. $\tau = 1$ and $\tau = 5$ were the most suitable – in a typical discussion in a chat line, the participants' on-line contributions only depend on a few previous comments which in our data are recorded in the preceding text windows. AR models of order >1 did not bring substantial improvement in the results; also, estimating an AR(1) model is computationally much simpler than more complex AR models.

The choice of the nonlinearity g in Formula (7) is another issue. The best results were obtained when g was chosen as $g(u) = \tanh(u)$, corresponding to imposing a 'cosh' prior on the residuals $s(t) - \alpha s(t - \tau)$. We have also previously [5] had good results with the simple $g(u) = \text{sign}(u)$ nonlinearity that corresponds to a Laplace prior on the residuals. In the ICA of static text documents, a nonlinearity $g(u) = u^2$ has been found successful in e.g. [15], corresponding to the skewed distribution of terms in documents. For dynamical text data, $g(u) = \tanh(u)$ was

[4]In a recent paper, Kolenda et al. [21] give a Bayesian method for choosing the number of estimated topics. We became aware of their work during the review process of this paper.

nevertheless better. Also, choosing a linear g (which neglects the non-gaussian, higher-order structure of the data) did not prove successful in our experiments.

4.2. RESULTS ON CHAT LINE DATA

The LSI of order $K = 100$ was computed as a preprocessing step as described in (2). Smaller K would also suffice, as we will demonstrate on another data set in the next section. We estimated $M = 10$ topics of chat line discussion simultaneously, using the orthogonal decorrelation presented in the end of Section 3. Figure 1 shows how different topic time series $\mathbf{w}_i^T\mathbf{Z}$, $i = 1, \ldots, M$ are activated at different times. We can see that the topics clearly are autocorrelated in time. The time span of Figure 1 is almost 24 hr; some topics are more or less persistent during the whole period and some will come up again after a few hours. The same fact can also be seen in the original text stream.

We now turn to analyze the projections $\mathbf{w}_i^T\mathbf{Z}_{\text{term}}$ of the terms onto minimum complexity directions. This information is complementary to that revealed by analyzing the document projections $\mathbf{w}_i^T\mathbf{Z}$, and offers an informative way of visualizing the results. By listing the terms corresponding to the highest values of $\mathbf{w}_i^T\mathbf{Z}_{\text{term}}$ we get a list of keywords for the i-th topic. The keywords are listed in Table I in the order of decreasing importance. It is seen that each keyword list indeed characterizes one distinct topic quite clearly. Due to polysemy, the same word may appear in more than one topic. Topic 1 deals with Jesse Jackson and his illegitimate child, topic 2 is about parental control over children's web usage and topic 3 is a general discussion about G. W. Bush. Topic 4 is a religious discussion, topic 5 deals with problems of the youth such as violence and drug abuse, and topic 6 is about the controversial flag of the state of Georgia, US, due to which the NCAA basketball games risked

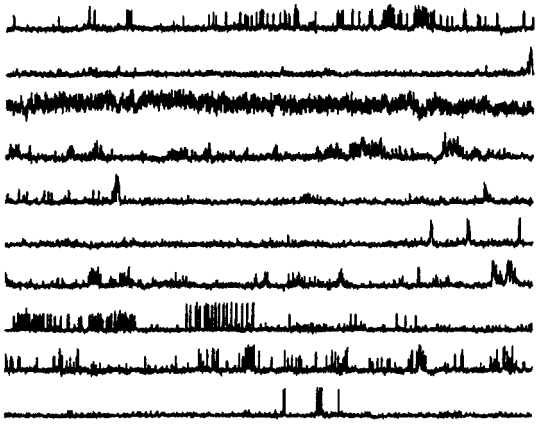

Figure 1. Activity of topics (vertical axis) in each chat window (horizontal axis). $g(u) = \tanh(u)$ and $\tau = 5$ were used in Formula (7). The uppermost time series corresponds to topic 1, the second to topic 2 etc.

Table I. Keywords of chat line discussion topics related to the time series in Figure 1.

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
jackson	site	bush	religion	violenc
sharpton	web	ashcroft	god	report
child	net	vote	jesu	youth
stori	word	kennedi	bibl	children
drudg	parent	presid	religi	gun
rainbow	nanni	cnn	life	point
monei	internet	time	follow	home
mistress	block	gore	read	drug
coalition	kid	question	stori	famili
tonight	system	elect	univers	satcher
pregnant	access	god	exist	health
affair	child	senat	faith	risk
black	base	power	man	factor
chenei	chat	thing	book	surgeon
jessi	page	fact	earth	prevent
Topic 6	Topic 7	Topic 8	Topic 9	Topic 10
flag	california	join	tax	free
move	power	discuss	cut	liber
citi	electr	est	exempt	opinion
ncaa	energi	tonight	monei	religion
offici	blackout	room	gop	form
atlanta	state	studio	hous	polit
count	deregul	cnn	congress	conserv
game	compani	conserv	pay	birth
night	crisi	american	interest	philosophi
georgia	price	nea	recess	establish
chang	plant	union	payer	narrow
lose	util	keen	secur	restrict
confeder	order	type	henri	independ
hehe	home	chat	hypocrit	orthodox
chenei	cost	newsroom	hyde	bound

cancellation in Atlanta. Topic 7 involves the energy shortage in California, topic 8 corresponds to comments given by the chat line moderator, topic 9 is about taxation and topic 10 is a short discussion dealing with the values of the politicians in the US.

One can compare the activities of the topic time series in Figure 1, and the term by document matrix frequencies of the first few keywords of each topic; the frequencies of the keywords nicely follow the activities of the time series.

The choice of the number of estimated topics is somewhat flexible. For example, estimating $M = 6$ topics would have given keyword lists similar to topics 2, 3, 4, 5, 6 and 7 in Table I.

The evaluation of the results based on the keywords is rather subjective. Numerical measures are hard to find as the chat line discussion data is not labeled. For this reason we present results on labeled data in the next section.

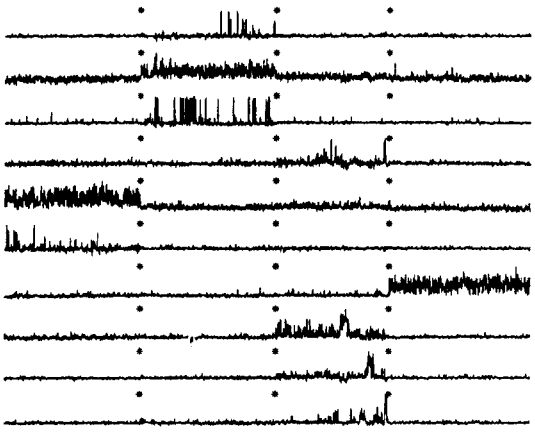

Figure 2. Activity of topics (vertical axis) in each newsgroup window (horizontal axis). $g(u) = \tanh(u)$ and $\tau = 5$ were used in Formula (7). The asterisks denote the newsgroup borders: sci.crypt, sci.med, sci.space and soc.religion.christian. The uppermost time series corresponds to topic 1, the second to topic 2 etc.

4.3. RESULTS ON NEWSGROUP DATA

In this section we present experimental results on newsgroup data where consecutive newsgroup articles are divided into overlapping windows similarly to what was done with the chat line data. Newsgroup data is often similar to chat line data in the sense that subsequent articles share a vague topic and the topic changes in time. The newsgroup data is labeled (as articles are from distinct newsgroups) and so we are able to quantitatively assess the separation results obtained by our algorithm and some other methods. The data is from four newsgroups of the 20 Newsgroup corpus[5]: sci.crypt, sci.med, sci.space and soc.religion.christian. The newsgroup articles, about 1000 from each group, were split to windows of 20 rows (excluding the headers) with 50% overlap between neighboring windows. Again, a binary representation of the documents was chosen but this time no stemming was used as newsgroup language tends to be quite precise, in contrast to chat line discussions. The size of the data matrix $\mathbf{X}$ was 5000 terms by 4695 documents.

LSI (2) of order $K = 50$ was computed as a preprocessing step. 6, 8 or 10 minimum-complexity directions $\mathbf{w}$ were estimated – discussion in a newsgroup can well be divided into subgroups, if more than one topic is dealt with. Figure 2 shows the topic time series $\mathbf{w}^T\boldsymbol{Z}$ in the case of 10 estimated topics. The asterisks in Figure 2 denote the borders between different newsgroups. It can be seen that each estimated topic time series corresponds to one of the newsgroups, or part of it. The keywords are seen in Table II, and they also nicely correspond to newsgroup labels: topics 1, 2 and 3 characterize different aspects discussed in sci.med, topics 4, 8, 9 and 10 in sci.space, topics 5 and 6 in sci.crypt and topic 7 is the only topic from soc.religion.christian.

[5]http://www.cs.cmu.edu/~textlearning

Table II. Keywords of newsgroup topics related to the time series in Figure 2.

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
human	problem	bank	design	kei
effect	diseas	skeptic	power	chip
food	scienc	intellect	station	govern
studi	medic	chastiti	control	encrypt
brain	studi	surrend	shuttl	secur
glutam	result	shame	orbit	clipper
review	food	won	option	public
level	effect	patient	human	system
singl	treatment	mar	provid	algorithm
paper	lot	medic	flight	david
diet	test	blood	engin	bit
industri	doctor	pittsburgh	modul	phone
blood	patient	comput	capabl	data
real	experi	practic	addition	nsa
high	medicin	migrain	system	escrow
Topic 6	Topic 7	Topic 8	Topic 9	Topic 10
phone	god	space	earth	matter
drug	christian	launch	venu	burst
commun	church	satellit	soviet	rememb
kei	christ	market	planet	star
life	sin	project	probe	black
dealer	jesu	commerci	mission	galaxi
assum	bibl	servic	surfac	red
crimin	approv	plan	mile	grb
discov	scriptur	orbit	kilomet	dark
hold	lord	cost	atmospher	gamma
motiv	homosexu	vehicl	venera	galact
terrorist	arami	note	lander	shift
compromis	faith	develop	orbit	object
system	love	fund	craft	show
polic	paul	nasa	balloon	energi

The classification error of the newsgroup documents is computed in the following way: The topic time series $\mathbf{w}_i^T \boldsymbol{Z}$ are first normalized to unit variance. Then a time series is mapped to the newsgroup whose documents have the highest sum of time series values in this particular time series. This is done at each time series separately. Now on the other hand, each document t is classified to that topic time series i in which the document projection $\mathbf{w}_i^T \boldsymbol{Z}(t)$ attains the maximum value. If the document is classified to a time series representing a different newsgroup than where the document was taken from, we consider the document misclassified. The total error is the percentage of misclassifications.

The results are seen in Table III which shows average results over 20 trials with different initial values for $\mathbf{w}$. Complexity pursuit is compared to ordinary ICA (FastICA [10]; this corresponds to complexity pursuit without the autoregressive

Table III. Results of estimating 10, 8 or 6 topics on dynamical text document data (newsgroup data) using complexity pursuit (with $g = \tanh$), JADE$_{TD}$ [24], ordinary FastICA (with $g = \tanh$), delayed decorrelation [20] and temporal predictability maximization [30]. Average results over 20 independent trials with different initial values for $\mathbf{w}$.

Method	Error $M = 10$	Flops $\cdot 10^9$	Error $M = 8$	Flops $\cdot 10^9$	Error $M = 6$	Flops $\cdot 10^9$
Compl. purs. $\tau = 1$	0.1515	9.29	0.1230	8.48	0.1081	8.01
Compl. purs. $\tau = 5$	0.1495	8.33	0.1423	7.82	0.1922	7.57
Compl. purs. $\tau = 10$	0.1737	8.27	0.1933	8.05	0.2760	7.53
JADE$_{TD}$ $\tau = 1$	0.1774	0.69	0.2043	0.55	0.2204	0.37
JADE$_{TD}$ $\tau = 5$	0.1774	0.79	0.2043	0.55	0.2204	0.37
JADE$_{TD}$ $\tau = 10$	0.1774	0.69	0.2043	0.55	0.2204	0.39
FastICA	0.4905	7.40	0.5460	7.16	0.6083	6.92
Del. decorr. $\tau = 1$	0.6591	1.38	0.6603	1.08	0.6920	0.77
Del. decorr. $\tau = 5$	0.6356	1.40	0.6700	1.08	0.6709	0.78
Del. decorr. $\tau = 10$	0.6688	1.38	0.6675	1.10	0.6852	0.77
Temp. pred. maxim.	0.4843	6.82	0.5442	6.82	0.6116	6.81

modeling of $s(t)$), JADE$_{TD}$ [24], Kolenda's delayed decorrelation [20] and Stone's temporal predictability maximization [30]. All these methods except ordinary ICA and the temporal predictability maximization consider the data at the current time instant and at some time lag τ; we present here results on $\tau = 1$, 5 and 10. The temporal predictability maximization instead considers short-time and long-time fluctuations in the data simultaneously.

As seen in Table III, complexity pursuit yields the smallest error of classification. Ordinary ICA, delayed decorrelation and temporal predictability maximization are not as successful as complexity pursuit and JADE$_{TD}$, giving evidence that both the temporal structure and information-theoretic measures of the data need to be taken into account. In all methods except JADE$_{TD}$ and delayed decorrelation, the data matrix is first reduced to $K = 50$ dimensions using LSI (SVD) and then $M = 10$, 8 or 6 topics are estimated. In the cases of JADE$_{TD}$ and delayed decorrelation, the LSI of order $K = M$ was computed in the beginning. This makes these two methods computationally less demanding than the other methods, as seen in Table III where the number of Matlab's floating point operations is given.

A new paper by Kolenda et al. [21] gives a method for determining the optimal lag parameter τ; this method is not applied here. The values for τ found in [21] are somewhat larger (naturally, this is data dependent) than those used in Table III, but testing e.g. values of $\tau = 20$, 50 or 100 in the delayed decorrelation method did not give any improvements on the results.

Figure 3 is an example of a box plot of the results, showing the variation in the results between different runs of the algorithms. All methods except JADE$_{TD}$ are sensitive to the initial choice of the vectors $\mathbf{w}$.

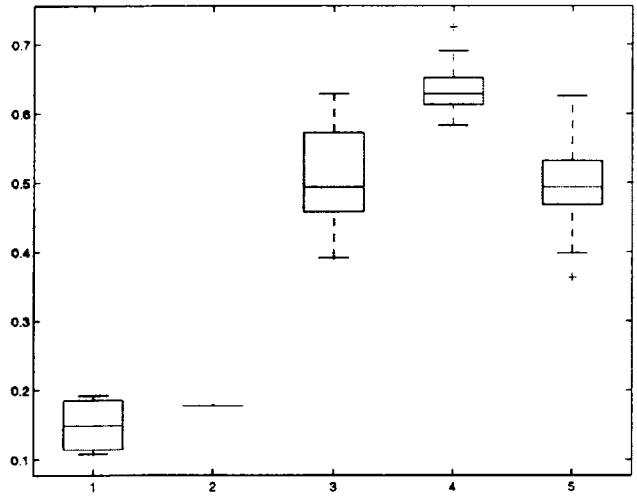

Figure 3. Box plot of the error in the case of $M = 10$ estimated topics and lag parameter $\tau = 5$. Methods from left to right: complexity pursuit, JADE$_{TD}$, ordinary FastICA, delayed decorrelation and temporal predictability maximization.

5. Conclusions

In this paper we have shown experimental results on how independent minimum complexity projections of a dynamic textual data identify some underlying latent or hidden topics in a dynamically evolving text stream. As an example of such dynamically evolving data we used chat line discussions. The method we used for finding the latent topics, complexity pursuit [11], is a generalization of projection pursuit to time series and amounts to estimating projections of the data whose approximative Kolmogoroff complexity is minimized. In our experiments the complexity pursuit algorithm was able to find distinct and meaningful topics of the discussion. We compared the complexity pursuit method to ordinary ICA and to ICA-type methods for time-dependent data: JADE$_{TD}$ [24], delayed decorrelation [20] and temporal predictability maximization [30]. In order to obtain numerical results we used labeled dynamical newsgroup data; complexity pursuit was the most successful in recognizing topically different newsgroup articles. Our results suggest that the method could serve in queries on temporally changing text streams, e.g. complementing other topic segmentation and tracking methods [1].

Acknowledgements

The authors are grateful to Thomas Kolenda for sharing his comments on the problem, and to Prof. Mikko Kurimo and the anonymous reviewers for giving valuable comments on the manuscript. E. Bingham has been partly supported by Ella and Georg Ehrnrooth Foundation and A. Kabán and M. Girolami have been partly supported by the Finnish National Technology Agency TEKES.

References

1. Allan, J., Carbonell, J., Doddington, G., Yamron, J. and Yang, Y.: Topic detection and tracking pilot study. Final report, In: *Proc. of DARPA Broadcast News Transcription and Understanding Workshop*, 1998, pp. 194–218.
2. Baeza-Yates, R. A. and Ribeiro-Neto, B.: *Modern Information Retrieval*, New York: ACM Press, 1999.
3. Belouchrani, A., Meraim, K. A., Cardoso, J.-F. and Moulines, E.: A blind source separation technique based on second order statistics, *IEEE Tr. on Signal Processing*, **45**(2) (1997), 434–444.
4. Berry, M. W., Dumais, S. T. and Letsche, T. A.: Computational methods for intelligent information access, In: *Proc. of Supercomputing '95*, San Diego, CA: USA, 1995.
5. Bingham, E., Kabán, A. and Girolami, M.: Finding topics in dynamical text: application to chat line discussions, In: *10th Int. World Wide Web Conf. Poster Proc.*, 2001, pp. 198–199.
6. Comon, P.: Independent component analysis—a new concept? *Signal Processing*, **36** (1994), 287–314.
7. Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K. and Harshman, R.: Indexing by latent semantic analysis, *Journal of the American Society for Information Science*, **41**(6) (1990), 391–407.
8. Friedman, J. H. and Tukey, J. W.: A projection pursuit algorithm for exploratory data analysis, *IEEE Tr. of Computers*, **c-23**(9) (1974), 881–890.
9. Hofmann, T.: Probabilistic Latent Semantic Analysis, In: *Proc. 15th Annual Conf. on Uncertainty in Artificial Intelligence (UAI'99)*, Sweden: Stockholm, 1999.
10. Hyvärinen, A.: Fast and robust fixed-point algorithms for independent component analysis, *IEEE Tr. on Neural Networks*, **10**(3) (1999), 626–634.
11. Hyvärinen, A.: Complexity pursuit: separating interesting components from time-series, *Neural Computation*, **13**(4) (2001), 883–898.
12. Hyvärinen, A., Karhunen, J. and Oja, E.: *Independent component analysis*, Wiley Interscience, 2001.
13. Isbell, C. L. and Viola, P.: Restucturing sparse high dimensional data for effective retrieval, In: *Advances in Neural Information Processing Systems 11*, 1998, pp. 480–486.
14. Jutten, C. and Herault, J.: Blind separation of sources, part I: An adaptive algorithm based on neuromimetic architecture, *Signal Processing*, **24** (1991), 1–10.
15. Kabán, A. and Girolami, M.: Unsupervised topic separation and keyword identification in document collections: a projection approach, Technical Report 10, Dept. of Computing and Information Systems, Univ. of Paisley, 2000.
16. Kabán, A. and Girolami, M.: A combined latent class and trait model for the analysis and visualization of discrete data, *IEEE Tr. on Pattern Analysis*, **23**(8) (2001), 859–872.
17. Kabán, A. and Girolami, M.: A dynamic probabilistic model to visualize topic evolution in text streams, *Journal of Intelligent Information Systems, Special Issue on Automated Text Categorization*, **18**(2) (2002).
18. Katz, S.: Distribution of content words and phrases in text and language modeling, *Natural Language Engineering*, **2**(1) (1996), 15–59.
19. Kohonen, T., Kaski, S., Lagus, K., Salojärvi, J., Honkela, J., Paatero, V. and Saarela, A.: Self organization of a massive document collection, *IEEE Tr. on Neural Networks*, **11**(3) (2000) 574–585. Special Issue on Neural Networks for Data Mining and Knowledge Discovery.
20. Kolenda, T. and Hansen, L. K.: *Dynamical components of chat*, Technical report Technical University of Denmark, 2000.

21. Kolenda, T., Hansen, L. K. and Larsen, J.: Signal detection using ICA: application to chat room topic spotting, In: Lee and Jung and Makeig and Sejnowski (eds.): *Proc. of the Third International Conference on Independent Component Analysis and Signal Separation (ICA2001)*, San Diego, CA: USA pp. 540–545, 2001.
22. Kolenda, T., Hansen, L. K. and Sigurdsson, S.: Independent components in text, In: M. Girolami (ed.): *Advances in Independent Component Analysis*, Springer-Verlag, 2000, Chapt. 13, pp. 235–256.
23. Molgedey, L. and Schuster, H. G.: Separation of a mixture of independent signals using time delayed correlations, *Phys. Review Letters,* **72**(23) (1994), 3634–3637.
24. Müller, K.-R., Philips, P. and Ziehe, A.: JADE$_{TD}$: Combining higher-order statistics and temporal information for blind source separation (with noise), In: *Proc. Int. Workshop on Independent Component Analysis and Signal Separation (ICA'99)*, France: Aussois, 1999, pp. 87–92.
25. Pajunen, P.: Blind source separation using algorithmic information theory, *Neurocomputing*, **22** (1998), 35–48.
26. Pajunen, P.: Blind source separation of natural signals based on approximate complexity minimization, In: *Proc. Int. Workshop on Independent Component Analysis and Signal Separation (ICA'99)*, France: Aussois, 1999, pp. 267–270.
27. Papadimitriou, C., Raghavan, P., Tamaki, H. and Vempala, S.: Latent semantic indexing: a probabilistic analysis, In: *Proc. 17th ACM Symp. Principles of Database Systems*, Seattle, 1998, pp. 159–168.
28. Salton, G. and McGill, M.J.: *Introduction to modern information retrieval*, New York: McGraw-Hill, 1983.
29. Slaney, M. and Ponceleon, D.: Hierarchical segmentation: finding changes in a text signal, In: *Proc. of the SIAM Text Mining 2001 Workshop*, Chicago, IL: 2001, pp. 6–13.
30. Stone, J. V.: Blind source separation using temporal predictability, *Neural Computation*, **13**(4) (2001).

PUBLICATION 5

Ella Bingham, Heikki Mannila, and Jouni K. Seppänen. Topics in 0-1 data. In David Hand, Daniel Keim, and Raymond Ng, editors, *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 450–455, Edmonton, Alberta, Canada, July 2002.

Topics in 0-1 Data

Ella Bingham
Ella.Bingham@hut.fi

Heikki Mannila
Heikki.Mannila@hut.fi

Jouni K. Seppänen
Jouni.Seppanen@hut.fi

Helsinki University of Technology
Laboratory of Computer and Information Science and HIIT Basic Research Unit
P.O. Box 5400, FIN-02015 HUT, Finland

ABSTRACT

Large 0-1 datasets arise in various applications, such as market basket analysis and information retrieval. We concentrate on the study of topic models, aiming at results which indicate why certain methods succeed or fail. We describe simple algorithms for finding topic models from 0-1 data. We give theoretical results showing that the algorithms can discover the epsilon-separable topic models of Papadimitriou et al. We present empirical results showing that the algorithms find natural topics in real-world data sets. We also briefly discuss the connections to matrix approaches, including nonnegative matrix factorization and independent component analysis.

Categories and Subject Descriptors

G.3 [**Probability and Statistics**]: Contingency table analysis; H.2.8 [**Database Management**]: Database Applications—*Data mining*; I.5.1 [**Pattern Recognition**]: Models—*Structural*

General Terms

Algorithms, Theory

1. INTRODUCTION

Large 0-1 datasets occur in various applications, such as market basket analysis, information retrieval, and mobile service use analysis. Lots of research has been done in the data mining community on methods for analyzing such data sets. The techniques can be roughly divided into two classes: (i) the methods based on frequent sets and association rules, aiming at discovery of interesting patterns, and (ii) probabilistic modeling methods aimed at discovering global structure from the data set.

In this paper we consider methods that fall in between these classes of methods. We study the identification of *topics* from 0-1 data. Intuitively, a topic is a set of interconnected variables such that, the occurrence value 1 in one of them tends to increase the probability of seeing value 1 for the other variables. The term "topic" comes from information retrieval: if a document concerns a certain topic, then the occurrence of some words is more probable than in the case when the document does not concern that topic. A single document can discuss many topics, and all words belonging to a topic need not appear in a document about that topic.

The concept of a topic is not restricted to document data. For example, in market basket data one can consider the customers having different topics in mind when they enter the store. A customer might for example want to purchase beer; the actual brand is selected only later, and perhaps she/he buys more than one brand.

The problem of finding topics in data has been considered using various approaches. Examples of the approaches are identification of finite mixtures, latent semantic indexing, probabilistic latent semantic indexing, nonnegative matrix factorization, and independent component analysis (see, e.g., [5, 10, 8, 14, 11, 13, 12]). Related work is considered in some more detail in Section 6.

We describe a simple topic model, corresponding to a generative model of the observations. The model states that there is a number of topics in the data, and that the occurrences of topics are independent. Given that the topic occurs, the words belonging to that topic are also considered to be independent. Later, we consider an extension of the model where the probabilities of topics vary from document to document (as in, e.g., [11, 14]).

The first question to address is whether actual data sets can be considered to be results of such generative process. Our definition of topic models implies, e.g., that negative correlations between variables are absent. We show that this is indeed the case on real data sets: while there are negative correlations, they are typically quite weak and cannot be considered to be violations of the model.

Given the class of topic models, the problem is then whether the model parameters can be estimated from the data. Our model class is close to the class of finite mixtures of multivariate Bernoulli distributions, a nonidentifiable class [10]. However, while in Bernoulli distributions the information obtained from 0 and 1 are on an equal footing, in our model the values 0 and 1 are not symmetric. This implies that for models where the topics are almost disjoint (e.g., the ε-separability condition of Papadimitriou et al. [14]) we can efficiently identify the topics and their parameters. Our main focus is whether there are some simple theoretical arguments that imply that simple topic models can be estimated from

SIGKDD '02 Edmonton, Alberta, Canada

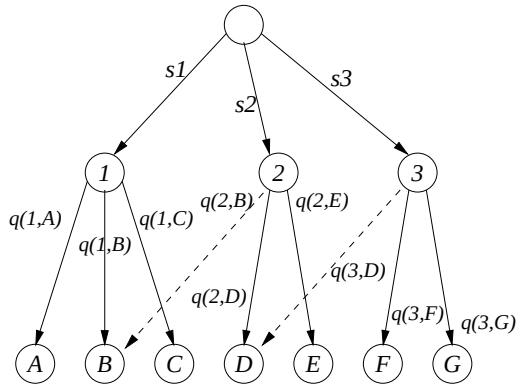

Figure 1: An example topic model

the data. We are able to show some first results in this direction, and support the results with empirical evidence.

The rest of this paper is organized as follows. In Section 2 we define the variants of topic models we consider. Section 3 describes the algorithms we use to find the topics. The theoretical results showing why the algorithms have a chance of working are given in Section 4. Some empirical results are described in Section 5. Related work is discussed in Section 6. Section 7 is a a short conclusion.

2. TOPIC MODELS

In this section we first introduce the simple topic model we consider, and then give an extension that corresponds to the model in [14]. We sometimes use the terminology of information retrieval, talking about documents instead of observations.

Given a set U of attributes, a *k-topic model* $\mathcal{T} = (\bar{s}, q)$ consists of k *topic probabilities* $\bar{s} = (s_1, \ldots, s_k)$ and a *topic-attribute probability matrix* q, giving for each $i = 1, \ldots, k$ and $A \in U$ an *attribute probability* $q(i, A)$ of A in topic i.

A document is sampled from $\mathcal{T}$ as follows. First, one selects independently which topics are on: topic i is on with probability s_i. All attributes are initially assigned value 0. Then, for each topic i that was selected, attribute A is assigned value 1 with probability $q(i, A)$.

Given a topic model $\mathcal{T} = (\bar{s}, q)$, the *weight* $w(i)$ of topic i in $\mathcal{T}$ is $\sum_{A \in U} q(i, A)$, i.e., the expected value of ones generated by topic i.

A topic model $\mathcal{T} = (\bar{s}, q)$ is *ε-separable*, if for each topic i there exists a set $U_i \subseteq U$ of attributes such that $U_i \cap U_j = \emptyset$ for $i \neq j$ and $\sum_{A \notin U_i} q(i, A) \leq \varepsilon w(i)$. That is, each topic i concentrates most of its mass to entries in U_i, and the overlap between these sets gets at most mass ε. We call U_i the *primary set of attributes* of topic i, and for $A \in U_i$ we say that i is the topic of A. If $A, B \in U_i$ for some i, we say that A and B belong to the same topic. Thus a 0-separable topic model is one where for each attribute A there is at most one topic i such that $q(i, A) > 0$.

Figure 1 illustrates an ε-separable topic model. The attribute set is $U = \{A, B, \ldots, G\}$, there are 3 topics, and the attribute subsets corresponding to the topics are $U_1 = \{A, B, C\}$, $U_2 = \{D, E\}$, and $U_3 = \{F, G\}$. The dashed arrows are examples of relationships that are possible in an ε-separable model with $\varepsilon > 0$.

A possible drawback with the above model for the generation of observations is that the topic probabilities s_i are considered to be constant: this could be considered unrealistic. Next we describe a variant, the *varying-probability topic model* in which they are also allowed to vary. Such a topic model is described as $\mathcal{T} = (\mathcal{S}, q)$, where $\mathcal{S}$ is a finite set of topic probability vectors $\bar{s}$.

A document is sampled from a varying-probability topic model by sampling first the topic probabilities $\bar{s}$ from $\mathcal{S}$, and then using the resulting topic model $(\bar{s}, q)$ as above. Thus this model is quite similar to the ones described in [14, 11]. The weight of a topic in such a model is defined to be the expected weight of topic under the sampling of the probability vector $\bar{s}$.

The condition of ε-separability is defined for varying-probability topic models in the same way as for normal topic models: at most a fraction of ε of the weight of each topic goes outside the primary attributes of that topic.

Given an 0-1 table over attributes U, denote for $A, B \in U$ by $p(A)$ the probability in the data of the event $A = 1$ and by $p(AB)$ the probability of $A = 1 \wedge B = 1$. Then the conditional probability $p(A|B)$ of A given B is of course $p(AB)/p(B)$. In practice, the probabilities are estimated as frequencies in the data.

There are certain degenerate cases in which the identification of topics does not succeed. For example, if there is one topic with one attribute, then different combinations of topic and attribute probabilities give the same observed frequency.

3. ALGORITHMS FOR FINDING TOPICS

In this section we describe two simple algorithms for finding topics. The first algorithm is applicable only to the basic model, while the second works also for varying-probability topic models.

Ratio algorithm. Consider first a k-topic 0-separable model $\mathcal{T} = (\bar{s}, q)$. Given two attributes A and B belonging to the same topic i, we have $p(A) = s_i q(i, A)$ and $p(B) = s_i q(i, B)$. Furthermore, $p(AB) = s_i q(i, A) q(i, B)$. Thus we have

$$\frac{p(A)p(B)}{p(AB)} = s_i.$$

If, however, A and B belong to different topics i and j, we have $p(A) = s_i q(i, A)$ and $p(B) = s_j q(j, B)$, and $p(AB) = s_i s_j q(i, A) q(j, B)$. Hence

$$\frac{p(A)p(B)}{p(AB)} = 1.$$

In the ε-separable case, any attribute may in principle be generated by any topic, and so $p(A) = \sum_i s_i q(i, A)$ and $p(AB) = \sum_i s_i q(i, A) q(i, B) + \sum_i \sum_{k \neq i} s_i s_k q(i, A) q(k, B)$.

Thus the algorithm for finding topics is simple. Compute the ratio $r(A, B) = p(A)p(B)/p(AB)$ for all pairs A and B; if the ratio is about 1, the attributes belong to different topics, if it is less than 1, the attributes might belong to the same topic.

Finding the topics from these ratios can be formalized as follows. We search for a partition of the set of attributes U into subsets so that within subsets most of the ratios $r(A, B)$ are close to a constant, and between subsets most of the ratios are close to 1. That is, given the matrix $r(A, B)$, where $A, B \in U$, and an integer k, find the partition of U to

subsets U_i for $i = 1, \ldots, k$, minimizing the score

$$\alpha \sum_{i=1}^{k} \sum_{A,B \in U_i} (r(A,B) - \gamma_i)^2 + \beta \sum_{i=1}^{k} \sum_{j=1,\ldots,k, j \neq i} \sum_{A \in U_i} \sum_{B \in U_j} (r(A,B) - 1)^2,$$

where α and β are constants and γ_i is the average of the ratios $r(A,B)$ within block U_i. This is a typical clustering problem, NP-complete in its general form, but lots of good approximate solutions exist.

This almost trivial method actually works quite nicely on some artificial and real data sets. However, it fails whenever the observations are generated using the varying-probability topic model. Thus we need more refined techniques.

Probe algorithm. Our second method is still quite simple. It is based on the method for finding similar attributes in 0-1 data described by Das et al. [7]. The basic intuition behind the algorithm is as follows. If two attributes A and B belong to the same topic, then the information that the occurrence of A (meaning the event $A = 1$) gives is about the same as the information given by the occurrence of B. Thus, if we have a measure for the similarity of the information given by two attributes, we can use that to find topics.

The *probe distance* $d(A,B)$ of two attributes is defined by

$$d(A,B) = \sum_{C \in U \setminus \{A,B\}} |p(C|A) - p(C|B)|.$$

The intuition here is that attributes A and B are similar if the distributions of the other attributes in the rows with $A = 1$ and in the rows with $B = 1$ are about the same. The attributes C serve as probes which are used to measure how similar the sets of rows are.

Our algorithm is as follows. Compute distances $d(A,B)$ for all pairs of attributes. (For a data set of n rows and p attributes, this can be done in time $O(np^2)$.) Again, find a partition of the set U of all attributes to subsets U_i minimizing the within-cluster distance and maximizing the distances between clusters. This can, of course, be solved using any clustering method. The details of the clustering are not our main focus; rather, we aim at giving results indicating why the method works. This is done in the next section.

4. PROPERTIES OF THE PROBE ALGORITHM

In this section we consider the properties of the probe algorithm given in the previous section. We first consider the case of 0-separable models, which naturally are quite simple. We show that for large sample sizes the distance between two attributes in the same topic tends to 0, and that the expected distance between two attributes belonging to different topics is quite large. We then consider the case of ε-separable models, and show that the same results continue to hold under some additional conditions. Most of the results are formulated under the assumption of no sample effects, i.e., by assuming infinite sample size.

We start with a lemma showing that for 0-separable models the distance between two attributes in the same topic goes to 0 as the sample size grows.

LEMMA 1. *Let r be a table of n rows over attributes U generated by a 0-separable topic model $\mathcal{T} = (\bar{s}, q)$. If A and B belong to the same topic U_i, then $\lim_{n\to\infty} d(A,B) = 0$.*

The next proposition extends this result to varying-probability topic models.

THEOREM 1. *Let r be a table of n rows over attributes U generated by a 0-separable varying-probability topic model $\mathcal{T} = (\mathcal{S}, q)$. Then, if A and B belong to the same topic U_i, then $\lim_{n\to\infty} d(A,B) = 0$.*

PROOF. Consider each probability vector $\bar{s} \in \mathcal{S}$. For the observations generated using the topic model $(\bar{s}, q)$ the lemma holds. As the statement of the lemma is independent of the actual topic probabilities s_i, the claim follows. □

LEMMA 2. *Let r be a table of n rows over attributes U generated by a 0-separable topic model $\mathcal{T} = (\bar{s}, q)$. If attribute A belongs to topic i, and attribute D belongs to topic j with $j \neq i$, then $E(d(A,D)) = (1 - s_i)(w(\mathcal{T}, i) - q(i,A)) + (1 - s_i)(w(\mathcal{T}, j) - q(j,D))$.*

THEOREM 2. *Let r be a table of n rows over attributes U generated by a 0-separable varying-probability topic model $\mathcal{T} = (\mathcal{S}, q)$. If attribute A belongs to topic i, and attribute D belongs to topic j with $j \neq i$, then $E(d(A,D)) = (1 - s_i)(w(\mathcal{T}, i) - q(i,A)) + (1 - s_i)(w(\mathcal{T}, j) - q(j,D))$.*

The proof is the same as for Theorem 1.

The above results show that the probe distances have a meaningful relationship to the topics of a 0-separable topic model. The details for general ε-separable models are far messier, but we give here an analogue of Lemma 1. The intuition is that when we add some weak links to a 0-separable model, the conditional probabilities are not perturbed too much, and thus the probe distances within a single topic will remain small. However, there are pathological ε-separable models: for example, consider a model where all attribute probabilities are much less than ε. Then, changes of the order of ε will naturally have a significant impact on the model. Of course, there is little hope of finding the topics in this kind of a model.

To rule out this kind of cases, there are several possibilities. For example, we can define the *distinctiveness* of an ε-separable topic model $\mathcal{T} = (\bar{s}, q)$ as the smallest value of the probability of an attribute being generated in the context of its primary topic:

$$\Delta(\mathcal{T}) = \min_{i, A \in U_i} s_i q(i,A),$$

where the minimum is taken over all topics i and all attributes $A \in U_i$. Thus, if a model has high distinctiveness ($\Delta(\mathcal{T}) \gg \varepsilon$), the generated attributes should usually reflect the topics they belong to.

An alternative restriction would be to say that the ε-separable topic model $\mathcal{T}$ has *θ-bounded conspiracy*, if for all attributes A with topic i we have $\sum_{j \neq i} q(j,A) \leq \theta$, i.e., the model $\mathcal{T}$ assigns at most a mass of θ to any attribute from topics other than its main topic. That is, the other topics do not conspire against a single attribute in a topic. Similar results as the one below can be proved for that case.

LEMMA 3. *Let r be a table of n rows over attributes U generated by a ε-separable topic model $\mathcal{T} = (\bar{s}, q)$. If attributes A and B belong to the same topic i, then $E(d(A,B)) \leq 2|U|k\varepsilon/\Delta(\mathcal{T})$.*

5. EMPIRICAL RESULTS

5.1 Experiments on simulated data

To evaluate how well do our algorithms perform, we generated artificial data according to our topic models described in Section 2. The data consisted of 100 attributes and 10 topics, each topic having a random number of primary attributes, and the number of observations was 100000. We performed tests on a ε-separable model with $\varepsilon = 0$, 0.01 and 0.1. In all experiments with the first (constant topic probabilities) model, the topic probabilities s_i were the same, so that we were able to test the effect of ε in model estimation accuracy.

Ratio algorithm. First we considered the ratios $r(A, B) = p(A)p(B)/p(AB)$. Recall that this should yield s_i, probability of topic i if A and B belong to the same topic i, and 1 otherwise, as then A and B are independent and their joint probability is separable. By listing these ratios in a matrix one can easily distinguish which topics belong to the same topic, as all of them have approximately the same ratio. In this way we can estimate the topic structure of the data, and also the topic probabilities s_i and topic-attribute probabilities $q(i, A)$ of A in topic i. Comparing to the true probabilities, the mean squared errors (MSE) of topic probabilities and the MSEs of topic-attribute probabilities are listed in Table 1 for $\varepsilon = 0$, 0.01 and 0.1. These figures are averages of 10 experiments. The variance between experiments was very small.

ε	MSE of topic probs.	MSE of topic-attr. probs.
0	$0.92 \cdot 10^{-4}$	$1.00 \cdot 10^{-3}$
0.01	$1.04 \cdot 10^{-4}$	$1.02 \cdot 10^{-3}$
0.1	$1.01 \cdot 10^{-4}$	$1.03 \cdot 10^{-3}$

Table 1: Mean squared errors of estimated topic and topic-attribute probabilities in the ratio algorithm.

In our varying-probability topic model, the topic probabilities s_i are randomly drawn for each document, and the ratio algorithm is not applicable.

Probe algorithm. Sammon mapping [17] is a convenient way to visualize how the attributes are grouped into distict topics. Figure 2 shows the Sammon map of the probe distances of the attributes in the 0-separable model. We can see that the attributes are nicely grouped into about 10 clusters, most of which are clear in shape. The clusters are not of equal size, as each topic has a random number of primary attributes. In the case of $\varepsilon = 0.01$, the clusters are a bit more vague in shape but still visible; with $\varepsilon = 0.1$, no clear clusters are seen anymore. The probe algorithm is quite resistant to the extension of varying topic probabilities: the Sammon maps are remarkably similar to those obtained for the nonvarying-probability topic models.

Maximum entropy model. We also considered whether the maximum entropy method described in e.g. [16, 15] might be useful in finding topics. The method is used to answer queries about the data as follows: first, one mines frequent sets with some threshold [1, 2], and then finds the maximum entropy distribution [3, 9] consistent with the frequent sets. We performed experiments using simulated data to see whether the results are consistent with the topic models used to generate the data. The results (not shown) indicate that this method does find results consistent with topic models quite satisfactorily but not perfectly. However, the performance is comparable only when the method is given roughly as much input as the simpler probe algorithm, and degrades badly when the frequency threshold increases and the input size decreases.

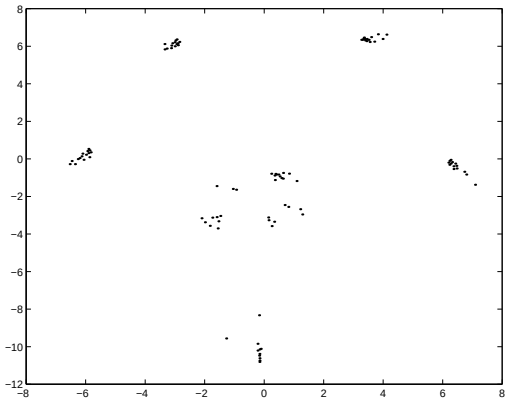

Figure 2: Sammon map of probe distances of attributes in artificial data; $\varepsilon = 0$.

5.2 Experiments on real data

Correlations. To determine the validity of the model assumptions on real data, we performed some trials on a collection of bibliographical data on computer science available on the WWW[1]. We call this the "Theory" dataset. As a preprocessing step, we removed all words occurring in fewer than 20 documents in the database. This reduced the number of words to 4227; the number of documents is 67066.

After preprocessing, we determined the probabilities $p(A)$ and $p(AB)$ for all words A, B (using word frequencies) and computed the covariances $\mathrm{cov}(A, B) = p(AB) - p(A)p(B)$. We can derive from the theoretical model in Section 2 that $\mathrm{cov}(A, B) \geq 0$ for all words A, B. This is not true in the dataset; indeed, more of the covariances are negative than positive. However, the distributions of the positive and negative covariances are very different. Figure 3 displays logarithmic histograms of the covariances in the Theory data. The histograms have been scaled to have equal areas. A short vertical line marks the position corresponding to one line in the database; covariances that are (absolutely) much smaller than this aren't usually very interesting, since they tend to reflect small-sample effects in cases where $p(AB)$ is very small (perhaps 0 or 1 lines) and $p(A)p(B)$ is nonzero but small.

Probe algorithm. We studied the behavior of the probe algorithm on the Theory bibliography. As a preprocessing step, we removed a small set of stop words and all numbers in the data, and then selected the 200 most frequent terms.

The probe distances of the terms were computed, and the term pairs with minimum probe distance are listed in Table 2. The table lists all pairs whose probe distance is under 1, in increasing order; the mean distance was about 2.7 and maximum distance about 6.2. The term pairs, most of which are abbreviations, are quite meaningful: e.g. 'stoc' is ACM Symp. on Theory of Computing and 'focs' is Symp.

[1] `http://liinwww.ira.uka.de/bibliography/Theory/Seiferas/`

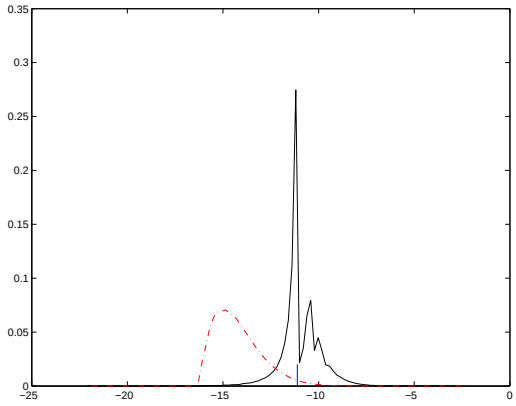

Figure 3: Histogram of $\ln(|\operatorname{cov}(A,B)|)$ for positive (solid) and negative (dashdotted) covariances for words A, B in Theory. A short vertical line marks $\ln(1/67066) = -11.1$.

dist.	terms		dist.	terms	
0.50	stoc	focs	0.91	jacm	libtr
0.63	infctrl	tcs	0.92	extended	abstract
0.63	tr	libtr	0.93	stacs	icalp
0.67	icalp	tcs	0.94	actainf	tcs
0.75	infctrl	icalp	0.95	fct	jcss
0.76	eurocrypt	crypto	0.95	fct	mfcs
0.79	mfcs	tcs	0.96	stacs	jcss
0.81	infctrl	jcss	0.96	jacm	tr
0.81	mfcs	icalp	0.96	sijdm	damath
0.81	jcss	tcs	0.97	ipps	jpdc
0.84	mfcs	infctrl	0.98	stoc	tr
0.86	mfcs	jcss	0.98	icpp	jpdc
0.88	jcss	icalp	0.99	sicomp	libtr
0.88	ipps	icpp	0.99	stacs	infctrl
0.89	mst	jcss	0.99	stacs	tcs

Table 2: Term pairs with minimum probe distance in the Theory data set

on Foundations of Computer Science; 'infctrl' is Information and Computation (formerly Information and Control) and 'tcs' is Theoretical Computer Science. For each term pair, the pair members belong to the same topical field, be it theoretical computer science, technical reports, cryptography, parallel processing, discrete mathematics etc. All these terms appear quite often in the data base, which makes the estimation of their probe distances reliable.

Does the method find topics? For example, listing the 10 terms with minimum probe distance to 'stoc' we get 'focs', 'tr', 'sicomp', 'libtr', 'stacs', 'jacm', 'jcss', 'icalp', 'infctrl', and 'ipl'. Computing the average distances of every term in this list to all other terms in the list, and taking the average of these averages, we get a distance of 1.17. On the other hand, computing the average distances of every term in this list to all other terms in the vocabulary, and again taking the average, yields 2.30. So the terms close to 'stoc' are also very close to one another but less close to other terms, and can thus be seen as forming a sort of topic. A similar comparison can be done to the closest neighbors of 'focs', giving a similar term list as above with similar average distances.

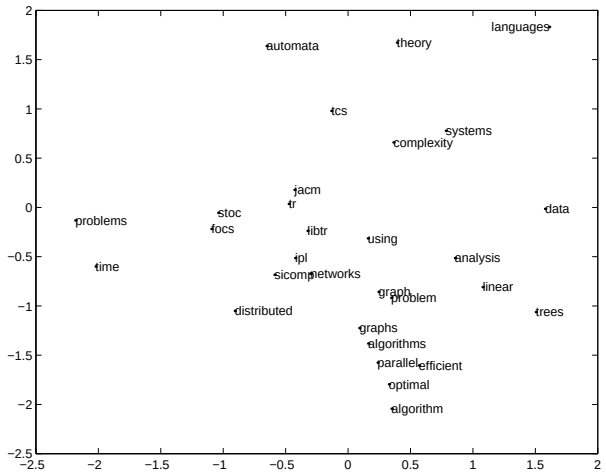

Figure 4: Sammon map of the probe distances of the 30 most common terms in the Theory data set.

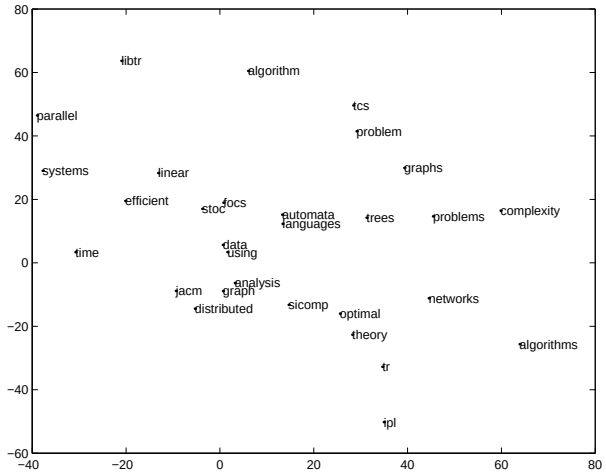

Figure 5: Sammon map of the LSI projections of the 30 most common terms in the Theory data set.

We used Sammon's mapping to project the data into two dimensions; Figure 4 shows how the 30 most common terms are located. There is clear evidence of clustering of related terms.

For comparison, we also projected the data into its 20-dimensional LSI [8] space. The Sammon map of the 30 most common terms is seen in Figure 5. In interpreting the figures, one should bear in mind that a two-dimensional Sammon map may not truly represent the locations of high-dimensional vectors.

6. RELATED WORK

The idea of looking at topics in 0-1 data (or other discrete data) has been considered in various contexts. The latent semantic indexing (LSI) method [8] uses singular-value decomposition (SVD) to obtain good choices of topics. This method works quite nicely in practice; the reason for this remains unclear. In a seminal paper [14], Papadimitriou et

al. gave some arguments justifying the performance of LSI. Their basic model is quite general and we have adopted their basic formalism; to obtain the results on LSI they have to restrict the documents to stem from a single topic. Of course, some restrictions are unavoidable.

Hofmann [11] has considered the case of probabilistic LSI. His formal model is close to ours, having the form $P(w|d) = \sum_z P(z|d)P(w|z)$, where the z's are topics, d refers to a document, and w to a word. Hofmann's main interest is in good estimation of all the parameters using the EM algorithm, while we are interested in having some reasoning explaining why the methods would find topics.

Cadez et al. [4] have considered the estimation of topic-like market-basket data, with the added complication that the same customer has multiple transactions, leading to the need of individual weights.

Our topic models are fairly close to the class of finite mixtures of multivariate Bernoulli distributions, a nonidentifiable class [10] (see also [5]). However, for those models, the values 0 and 1 have symmetric status, while for the topic models defined above this is not the case. We conjecture that the class of topic models is essentially identifiable provided that the topics are almost disjoint in, e.g., the ε-separability sense.

In nonnegative matrix factorization (NMF), an observed data matrix V is presented as a product of two unknown matrices: $V = WH$. All three matrices have nonnegative entries. Lee and Seung [13] give two practical algorithms for finding the matrices W and H given V. Restriction to binary variables is not straightforward in these algorithms.

Independent component analysis (ICA) [6, 12] is a statistical method that expresses a set of observed multidimensional sequences as a combination of unknown latent variables that are more or less statistically independent. Topic identification in 0-1 data can be interpreted in the ICA terminology as finding latent binary sequences, unions of which form the observed binary data. ICA in its original form relies heavily on matrix operations; for sparse data, union is roughly equivalent to summation, so methods for ICA could be considered for the problem at hand. Nevertheless, most existing ICA algorithms are suitable for continuosly distributed data with Gaussian noise — the case of 0-1 variables and Bernoulli noise is quite different, and practical ICA algorithms tend to fail in this case.

7. CONCLUSIONS

We have considered the problem of finding topics in 0-1 data. We gave a formal description of topic models, and showed that relatively simple algorithms can be used to find topics from data generated using such models. We showed that the probe algorithm works reasonably well in practice.

Lots of open issues remain, both on the theoretical and on the practical side. The detailed relationship of our model compared to, e.g., Hofmann's model remain to be studied. We conjecture that the topic models are identifiable, in contrast with general mixtures of multivariate Bernoulli distributions. Understanding the behavior of LSI is still open. Similarly, seeing how nonnegative matrix factorization is connected to the other approaches is open, as are the ways of extending ICA to the Bernoulli case.

8. REFERENCES

[1] R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. In *SIGMOD '93*, pages 207–216, 1993.

[2] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A. I. Verkamo. Fast discovery of association rules. In U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, editors, *Advances in Knowledge Discovery and Data Mining*, chapter 12, pages 307–328. AAAI Press, 1996.

[3] A. L. Berger, S. A. Della Pietra, and V. J. Della Pietra. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71, 1996.

[4] I. V. Cadez, P. Smyth, and H. Mannila. Probabilistic modeling of transaction data with applications to profiling, visualization, and prediction. In *KDD 2001*, pages 37–46, San Fransisco, CA, Aug. 2001.

[5] M. A. Carreira-Perpinan and S. Renals. Practical identifiability of finite mixtures of multivariate Bernoulli distributions. *Neural Computation*, 12:141–152, 2000.

[6] P. Comon. Independent component analysis — a new concept? *Signal Processing*, 36:287–314, 1994.

[7] G. Das, H. Mannila, and P. Ronkainen. Similarity of attributes by external probes. In *Knowledge Discovery and Data Mining*, pages 23–29, 1998.

[8] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407, 1990.

[9] S. Della Pietra, V. J. Della Pietra, and J. D. Lafferty. Inducing features of random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(4):380–393, 1997.

[10] M. Gyllenberg, T. Koski, E. Reilink, and M. Verlaan. Non-uniqueness in probabilistic numerical identification of bacteria. *Journal of Applied Probability*, 31:542–548, 1994.

[11] T. Hofmann. Probabilistic latent semantic indexing. In *SIGIR '99*, pages 50–57, Berkeley, CA, 1999.

[12] A. Hyvärinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. John Wiley & Sons, 2001.

[13] D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401:788–791, Oct. 1999.

[14] C. H. Papadimitriou, P. Raghavan, H. Tamaki, and S. Vempala. Latent semantic indexing: A probabilistic analysis. In *PODS '98*, pages 159–168, June 1998.

[15] D. Pavlov, H. Mannila, and P. Smyth. Probabilistic models for query approximation with large sparse binary datasets. In *UAI-2000*, 2000.

[16] D. Pavlov and P. Smyth. Probabilistic query models for transaction data. In *KDD 2001*, 2001.

[17] J. W. Sammon. A nonlinear mapping for data structure analysis. *IEEE Transactions on Computers*, 18(5):401–409, May 1969.

PUBLICATION 6

Jouni K. Seppänen, Ella Bingham, and Heikki Mannila. A simple algorithm for topic identification in 0-1 data. In Nada Lavrač, Dragan Gamberger, Ljupčo Todorovski, and Hendrik Blockeel, editors, *Knowledge Discovery in Databases: PKDD 2003. 7th European Conference on Principles and Practice of Knowledge Discovery in Databases. Cavtat-Dubrovnik, Croatia, September 2003, Proceedings*, number 2838 in Lecture Notes in Artificial Intelligence, pages 423–434. Springer, 2003.

A simple algorithm for topic identification in 0–1 data

Jouni K. Seppänen, Ella Bingham, and Heikki Mannila

Laboratory of Computer and Information Science and HIIT Basic Research Unit,
Helsinki University of Technology

Abstract. Topics in 0–1 datasets are sets of variables whose occurrences are positively connected together. Earlier, we described a simple generative topic model. In this paper we show that, given data produced by this model, the lift statistics of attributes can be described in matrix form. We use this result to obtain a simple algorithm for finding topics in 0–1 data. We also show that a problem related to the identification of topics is NP-hard. We give experimental results on the topic identification problem, both on generated and real data.

1 Introduction

Large collections of 0–1 data occur in many applications, such as information retrieval, web browsing, telecommunications, and market basket analysis. While the dimensionality of such data sets can be large, the variables (or attributes) are seldom completely independent. Rather, it is natural to assume that the attributes are organized into (possibly overlapping) *topics*, i.e., collections of variables whose occurrences are somehow connected to each other.[1] For example, in document data the topics correspond to topics of the document: e.g., phrases "data mining", "decision trees" and "association rules" probably are included in one topic, which might be called the "data mining" topic. In supermarket market basket data, the topics could correspond to classes of products such as soft drinks, vegetables, etc. In discretized gene expression data topics could correspond to groups of genes that are expressed in similar conditions or tissues.

Finding topics from data is by no means easy: the topics can be overlapping, and a particular topic is active only for a subset of documents. For example, simple frequent set based approaches are unable to find topics, as the attributes in a topic are seldom 1 together. There has been lots of work that searches for latent structure in 0–1 data (see, e.g., [1–10]). The approaches range from simple methods based on covariance-type statistics (e.g., [9]) to full probabilistic models (e.g., [4]) and to spectral approaches [10].

In order to discover topics from 0–1 data, one first has to specify the model for topics, and then give a method that finds topics corresponding to the model.

[1] Our usage of the word *topic* is similar but not identical to the meaning in information retrieval literature, where a topic is a probability distribution on the universe of terms, typically concentrating on a few terms.

N. Lavrač et al. (Eds.): PKDD 2003, LNAI 2838, pp. 423–434, 2003.

In this paper we describe a simple generative topic model, based on our previous work [11]. We prove some analytical results about the model by using the concept of lift [12]. We show that the lift statistics of individual attribute pairs can be described in matrix form as linear combinations of lift statistics of disjoint topics. Based on this observation, we give a simple algorithm for finding topics in 0–1 data. We also show that one form of the topic identification problem is NP-hard. We give experimental results on both generated and real data, showing that the algorithm works well in practice.

First we review some other methods for finding latent structure in binary data. Many of these generative models are quite powerful and are able to describe complex situations. On the other hand, finding exact solutions for them is computationally intractable, and it is difficult to get a clear picture of the quality of the obtained estimates. Many of the methods are also symmetric with respect to the data values 0 and 1; on the basis of the asymmetry in the data generating process, this can be viewed as a potential source of problems.

In nonnegative matrix factorization (NMF) [1], an observed data matrix is decomposed into a product of two unknown matrices. All three matrices have nonnegative entries. The observed data is regarded as a sum of latent variables. Lee and Seung give two algorithms for finding the unknown matrices; there is, however, no probabilistic interpretation of the results of NMF. Computationally, the methods seems very demanding and there are no clear results on the quality of the solutions [13].

The latent semantic analysis (LSA) method [2] uses singular-value decomposition to decompose an observed data matrix into a product of matrices. (In contrast to NMF, the matrices can have negative entries, too.) In a seminal paper by Papadimitriou et al. [3] some arguments were given to justify the performance of LSI by presenting a probabilistic corpus model. Their basic model is quite general and somewhat similar to ours.

Hofmann [4] has presented a probabilistic version of LSA, termed PLSA. His formal model is fairly close to ours and we will show comparative results on the models. For each observation vector, some topics are first selected according to some observation-specific topic probabilities; then, the topics generate attributes according to some topic-attribute probabilities. The attributes are conditionally independent given the topic. Hofmann's main interest is in good estimation of all the parameters using the EM algorithm, while we are interested in the structure of the data (that is, the probabilities of attributes belonging to topics) and also explaining why the methods would find topics.

Laten Dirichlet Allocation (LDA) [14–16] is a method in which the data model is closely similar to Hofmann's PLSA but the estimation of the parameters is computationally more demanding: a variational approximation to the data likelihood is needed prior to EM estimation of the parameters. Independent component analysis (ICA) ([8, 17, 18]) is a statistical method that expresses observed multidimensional sequences as combinations of unknown latent variables, that are statistically as independent as possible. The so called probe distances [19] of attributes can be used to find (possibly overlapping) sets of attributes that

behave similarly with respect to other attributes; we studied this in an earlier paper [11]. Cooley and Clifton [9] compute the frequent sets in the data and cluster them using a hypergraph partitioning scheme, thus avoiding the problem of not having all attributes of a topic present in one data vector.

A popular method to analyze 0–1 data is the class of finite mixtures of multivariate Bernoulli distributions. However, for the Bernoulli models, the values 0 and 1 have symmetric status, while for our topic models defined in Section 2 this is not the case. Another important difference between Bernoulli (or any other) mixture model and our model is that in mixture models it is assumed that an observed 0–1 vector is only generated by one latent topic, although generation probabilities are given for all latent topics. In this paper we assume that a data vector is generated by the interaction of several latent topics. Binary generative topographic mapping [20, 21] also assumes that the data vectors are generated by one latent topic at a time.

The rest of this paper is organized as follows. We describe our model and examine some of its analytical properties in Section 2. In Section 3 we study the lift statistic and describe the simple algorithm based on it. We give experimental results in Section 4, and conclude in Section 5.

2 Topic Models

In this section we present our concept of a topic model, give the likelihood function of the model, and discuss what kinds of parameter values are realistic. This form of the model was introduced earlier by us [11].

Let U be an n-element set of *attributes* (e.g., words). A *k-topic model* $\mathcal{T}$ arranges the n attributes into k *topics*. The model has the following parameters: a k-element vector $\boldsymbol{s} = (s_1, \ldots, s_k)$ corresponding to the k topics, and a $k \times n$ matrix $\mathbf{Q}$ whose elements relate the topics to the attributes; the element corresponding to topic i and attribute A is denoted by $Q_{i,A}$. All elements of $\boldsymbol{s}$ and $\mathbf{Q}$ must be probabilities, i.e., reals in the range $[0, 1]$; however, neither $\boldsymbol{s}$ nor any row or column of $\mathbf{Q}$ is required to sum up to 1.

A data vector $\boldsymbol{x}$ (e.g., a document) is sampled from $\mathcal{T}$ as follows. First, the active topics are selected by sampling a k-element binary vector $\boldsymbol{t}$ whose every component t_i is 1 with probability s_i, independently of all other components. Second, the active topics generate the attributes. For each topic i, an n-element binary vector $\boldsymbol{x}_i$ is sampled so that the component corresponding to A is 1 with probability $t_i Q_{i,A}$, independently of all other components. The data vector $\boldsymbol{x}$ is then the logical *or* (i.e., maximum) of all the vectors $\boldsymbol{x}_i$, $\boldsymbol{x} = \bigvee_{i=1}^{k} \boldsymbol{x}_i$.

It would be possible to add another layer on top of the topics, selecting the topic probabilities anew for each data vector from, e.g., a Dirichlet distribution. Many of our results could be generalized to such settings, which however fall outside the scope of this treatment. This type of approach has been taken in [3, 4, 14–16].

We next present the likelihood function of a k-topic model $\mathcal{T}$ with parameters $\boldsymbol{s}, \mathbf{Q}$. The data D consists of vectors $\boldsymbol{x}$, each considered independently of

the others,

$$P(D \mid \mathcal{T}) = \prod_{\boldsymbol{x} \in D} P(\boldsymbol{x} \mid \mathcal{T}).$$

The probability of a single observation $\boldsymbol{x}$ is

$$P(\boldsymbol{x} \mid \mathcal{T}) = \sum_{\boldsymbol{t}} P(\boldsymbol{t} \mid \mathcal{T}) P(\boldsymbol{x} \mid \boldsymbol{t}, \mathcal{T}).$$

The sum is taken over all k-element 0–1 vectors $\boldsymbol{t}$, corresponding to all 2^k possible combinations of active topics. The probability of a topic combination depends on the parameters $\boldsymbol{s}$ only,

$$P(\boldsymbol{t} \mid \mathcal{T}) = P(\boldsymbol{t} \mid \boldsymbol{s}) = \prod_{i=1}^{k} P(t_i \mid s_i) = \prod_{i=1}^{k} s_i^{t_i} (1 - s_i)^{1 - t_i}.$$

The probability of an observation given the active topics depends on the parameters $\mathbf{Q}$ only,

$$P(\boldsymbol{x} \mid \boldsymbol{t}, \mathcal{T}) = P(\boldsymbol{x} \mid \boldsymbol{t}, \mathbf{Q}) = \prod_{A \in U} P(x_A \mid \boldsymbol{t}, \mathbf{Q}),$$

where x_A denotes the element of $\boldsymbol{x}$ that corresponds to the attribute $A \in U$. A single attribute has a value of either zero or one, with distribution

$$P(x_A \mid \boldsymbol{t}, \mathbf{Q}) = p_A^{x_A} (1 - p_A)^{1 - x_A} = \begin{cases} 1 - p_A, & x_A = 0 \\ p_A, & x_A = 1, \end{cases}$$

where

$$p_A = 1 - \prod_{i=1}^{k} (1 - Q_{i,A})^{t_i}.$$

The likelihood function, if expanded fully, would have a large number of terms because of the sum over 2^k topic combinations $\boldsymbol{t}$. This suggests a high computational complexity, and indeed the task of selecting the best $\boldsymbol{t}$ is difficult. This is illustrated by the following theorem, whose proof we defer to the Appendix.

Theorem 1. *The following problem is* NP*-complete: given a topic model $\mathcal{T}$, a single data vector $\boldsymbol{x}$ and a threshold ρ, decide whether there is a topic assignment $\boldsymbol{t}$ such that the probability of the data given the assignment exceeds the threshold, $P(\boldsymbol{x} \mid \boldsymbol{t}, \mathcal{T}) \geq \rho$.*

However, the models involved in the proof would best be described as contrived, so the result should not dissuade us from researching some reasonable subclass of topic models. But what kind of models are reasonable?

One assumption that we will make is that the topic probabilities s_i are small. This seems reasonable at least in the context of document data: if some words occur in a large fraction of all documents, in information retrieval they would

be classified as stop words and not considered in searches; it is the less common words that distinguish interesting documents.

Another question is the amount of overlap between topics – if two topics consist of almost completely the same attributes, it does not seem easy to distinguish between them. In [11] we considered a class of "ϵ-separable" models, an idea similar to that in [3]. A model is ϵ-separable if every topic has a set of primary attributes and assigns at most a fraction ϵ of its attribute-activation weight to the non-primary attributes. However, the ϵ-separability property does not perfectly capture the idea of almost-disjoint topics, as the discussion in [11, before Lemma 3] notes: for example, several topics can "conspire" against another topic i by giving high weight to one of i's primary attributes. Even if every high weight is less than a fraction ϵ of the topic's total weight, it is possible that the majority of activations of that attribute come from the conspiring topics and not the primary topic.

This leads us to define a different separability concept: a model has *θ-bounded conspiracy* if every attribute A has a primary topic i such that

$$\sum_{j \neq i} Q_{j,A} \leq \theta Q_{i,A}.$$

We conjecture that a model is discoverable from data if it has low values of s_i and conspiracy bounded by some low θ.

3 Using the Lift Statistic

We now consider a statistic commonly called called *lift* or *interest* [12, 22, 23],

$$\text{lift}(A, B) = \frac{P(A \mid B)}{P(A)} = \frac{P(A, B)}{P(A)P(B)},$$

which is a kind of a relative risk factor: how much more common is it to observe A given that B is observed, compared to no information about B? Lift was chosen because it measures dependence, which is highly relevant to topic models – when two attributes belong strongly to the same topic, their co-occurrence should deviate significantly from the independence assumption. For independent A and B, $\text{lift}(A, B) = 1$, and the stronger the (positive) dependence, the higher the lift. Note that our model predicts $\text{lift}(A, B) \geq 1$ for all pairs $A, B \in U$; thus, one way of assessing whether the model fits a given data set is to see how $\text{lift}(A, B)$ is actually distributed.

Proposition 1. *Assume that attribute A is only generated by topic i. Then for any attribute B,*

$$\text{lift}(A, B) = \frac{P(t_i \mid B)}{P(t_i)} = \frac{P(t_i, B)}{P(t_i)P(B)}.$$

Proof. We factorize the probabilities: $P(A) = P(A, t_i) = P(t_i)P(A \mid t_i)$ and $P(A, B) = P(t_i, A, B) = P(t_i)P(B \mid t_i)P(A \mid t_i, B)$. Since A is only generated by topic i, $P(A \mid t_i, B) = P(A \mid t_i)$. Thus

$$\text{lift}(A, B) = \frac{P(A, B)}{P(A)P(B)} = \frac{P(t_i)P(A \mid t_i)P(B \mid t_i)}{P(t_i)P(A \mid t_i)P(B)}.$$

Using Bayes' theorem $P(B \mid t_i) = P(B)P(t_i \mid B)/P(t_i)$ and canceling terms we obtain the result. □

What Proposition 1 says is that if A is a "core attribute" of topic i, i.e., an attribute generated by i only, then A represents i perfectly in lift calculations, even if $Q_{i,A} < 1$. Of course in practice, when the lift must be estimated from data, a small value of $Q_{i,A}$ can cause poor results. Another point to note is that the probability $P(B \mid t_i)$ appearing in the proof is *not* the model parameter $Q_{i,B}$. Instead, it is the probability that any topic will generate B conditioned on the fact that at least topic i is active. Proposition 1 has as immediate consequences two results that we used already in [11].

Corollary 1. *If attributes A and B are only generated by topic i, i.e., $Q_{j,A} = Q_{j,B} = 0$ for $j \neq i$, then* $\text{lift}(A, B) = s_i^{-1}$.

Corollary 2. *If attribute A is only generated by topic i and attribute B is only generated by topic j, then* $\text{lift}(A, B) = 1$.

Thus, the lift statistic between attributes belonging to one topic only is very simple. The interesting question is how lift behaves when an attribute belongs to several topics.

Assume that attribute A is only generated by topic i, and attribute B is generated by both topics i and j. Now $\text{lift}(A, B)$ is, after simplification,

$$\frac{P(A, B)}{P(A)P(B)} = \frac{Q_{i,B} + s_j Q_{j,B} - Q_{i,B} s_j Q_{j,B}}{s_i Q_{i,B} + s_j Q_{j,B} - s_i s_j Q_{i,B} Q_{j,B}} \approx \frac{Q_{i,B} + s_j Q_{j,B}}{s_i Q_{i,B} + s_j Q_{j,B}}$$

where in the approximation we have assumed that $Q_{i,B} s_j Q_{j,B}$ and $s_i s_j Q_{i,B} Q_{j,B}$ are small compared to the other terms. The above formula generalizes to the case where B is generated by some other topics than i and j, too: before the approximation we then have several second order terms $s_\ell Q_{\ell,B}$ corresponding to all topics ℓ that generate B, and similarly several third order terms $s_\ell Q_{i,B} Q_{\ell,B}$ (in the numerator) or fourth order terms $s_i s_\ell Q_{i,B} Q_{\ell,B}$ (in the denominator).

Assume now that all the topic probabilities are (approximately) equal, i.e., $s_\ell \approx s$ for all topics ℓ. Then we can write the above formula as $\text{lift}(A, B) \approx (s^{-1} Q_{i,B} + Q_{j,B})/(Q_{i,B} + Q_{j,B})$. Furthermore, let each topic ℓ have c_ℓ core attributes that are only generated by that topic. Then using Corollaries 1 and 2 we note that the lifts of A and all core attributes can be included in the formula as follows:

Observation. The lift between a core attribute A of topic i and an attribute B generated by topics i and j is

$$\mathrm{lift}(A,B) \approx \sum_{A'} \mathrm{lift}(A,A')c_i^{-1}\frac{Q_{i,B}}{Q_{i,B}+Q_{j,B}} + \sum_{D'} \mathrm{lift}(A,D')c_j^{-1}\frac{Q_{j,B}}{Q_{i,B}+Q_{j,B}}$$

where $\sum_{A'} \mathrm{lift}(A,A')c_i^{-1}$ is an averaged estimate of s^{-1}, $\sum_{D'} \mathrm{lift}(A,D')c_j^{-1} = 1$ and the two sums run over the core attributes A' and D' of topics i and j, respectively. Also, we may add a third summation including $\mathrm{lift}(A,F')$ where F' is a core attribute belonging to topic l into which B does not belong to, as then $Q_{l,B} = 0$ and the whole term vanishes. This observation again generalizes to the case where B is generated by multiple topics.

The above reasoning included approximations in discarding high-order terms and the somewhat crude assumption that all s_i are equal. In any case, it does yield an idea of how to discover topics: for an attribute B that belongs to several topics, define a vector $\boldsymbol{\alpha}$ whose length is the total number of all core attributes. The element corresponding to A (a core attribute of topic i) is $\alpha_A = Q_{i,B}/(c_i \sum_j Q_{j,B})$. Then $\mathrm{lift}(A,B) \approx \boldsymbol{\alpha}^T \mathrm{lift}(A,\cdot)$ for all core attributes A, where we denote by $\mathrm{lift}(A,\cdot)$ the vector of lifts between A and all core attributes (where $\mathrm{lift}(A,A) = 0$). This gives us an algorithm for finding the topics in which the attributes belong, and also the parameters Q:

- Identify those attributes that belong to one topic only – this can be done by looking at the lift statistics, which are always either 1 or $1/s$ for those attributes.
- Cluster those attributes using some traditional clustering algorithm; at this stage the clusters do not overlap and do not cover all attributes – if an attribute B belongs to several topics, its lifts are intermediate between 1 and $1/s$, and so B is not clustered. For A belonging to one topic i only, $Q_{i,A} = P(AA')/P(A')$ which can be averaged over all A' belonging to the same topic i as A.
- For attributes B which are not clustered, find a decomposition $\mathrm{lift}(B,\cdot) = \boldsymbol{\alpha}^T R$, where the square symmetric matrix R has the vectors $\mathrm{lift}(A,\cdot)$ (of already clustered attributes A) as its columns. All of the lifts in this formula are known, so the vector $\boldsymbol{\alpha}$ can be estimated straightforwardly. The elements of $\boldsymbol{\alpha}$ are nonzero for those attributes that share a topic with B, and zero for others. Also, the elements are more or less constant within attributes of a given topic. Now $Q_{i,B} = \alpha_A c_i / \sum_j Q_{j,B}$ where α_A can be averaged over all A' belonging to topic i, c_i is known, and for small and equal s_j we can approximate $P(B) \approx s \sum_j Q_{j,B}$, which gives us $\sum_j Q_{j,B}$. We can also assume $\sum_j Q_{j,B} = 1$ and scale the estimated $Q_{i,B}$ accordingly.

4 Experimental Results

4.1 Generated Data

We designed experiments to see how the conspiracy statistic θ of a model affects our clustering results. The results corroborate our conjecture that low-

conspiracy models are easier to discover. We constructed random models with θ-bounded conspiracy using the following recipe. The model has 10 topics and 100 attributes. The probability s_i of a topic was drawn uniformly at random from the interval $[0.01, 0.5]$. Each attribute was assigned a primary topic so that each topic was primary for 10 attributes.

To assign the within-topic attribute probabilities $Q_{i,A}$ so that the conspiracy parameter is θ, we first drew a number p uniformly from $[0, 1]$ and let $Q_{i,A} = p$ for the primary topic i. Then we distributed the mass θp to the non-primary topics in an uneven way. Each non-primary topic in random order received a fraction of ϕ of the remaining mass, where ϕ is chosen at random from $[0, 1]$, separately for each non-primary topic. The last topic received all remaining mass to make the mass sum up exactly to θp.

This way of generating a random model includes a number of somewhat arbitrary choices that we now justify. First, the topic probabilities s_i were chosen not from $[0, 1]$ but from a smaller interval. Some lower limit is necessary so that each topic is represented in a finite data sample; and an upper limit is needed by our algorithm, which distinguishes a topic by estimating its probability and cannot discover a topic that is almost always active. In a preliminary test (not shown), our algorithm's performance was best with low upper limits, and deteriorated rapidly when the upper limit approached 1. We chose 0.5 as the upper limit as a conservative approach: in document data, one would expect that individual topics have much smaller probabilities.

Second, we discuss the distribution of the within-topic attribute probabilities of non-primary topics. A more obvious strategy would be to draw the probabilities independently and then to normalize, but then the distribution would have become more even. With 9 non-primary topics, all the probabilities would center around $\theta/9$ times the primary probability, which makes the task far easier: none of the non-primary topics is likely to be confused with the primary topic. In contrast, our procedure typically results in a few non-primary topics with non-negligible topic-attribute probabilities for each attribute. We wish to mimic the behavior of true data sets, such as text document data: a term may have several meanings, perhaps a primary meaning and one or few secondary meanings, hence it belongs primarily to one topic of discussion and secondarily to a few other topics, but not to all possible topics.

In the experiment, we estimated the topic-attribute probabilities $\mathbf{Q}$ using the lift statistic, NMF, PLSA[2] and K-means. The NMF and PLSA methods estimate $\mathbf{Q}$ given the observed binary data. A naive alternative is the simple K-means algorithm which clusters the attributes into non-overlapping sets; we assume that $Q_{i,A}$ is equal for all attributes A of topic i and sums to 1 at each topic.

Figure 1 shows the mean squared errors (MSE's) of the estimated $\mathbf{Q}$, compared to the true probabilities used to generate the data. The conspiracy parameter θ runs from 0 to 1. At each θ, the topic probabilities s are sampled anew, so there is great variability in the data models. Originally, the topic-attribute prob-

[2] The PLSA method was kindly programmed by Mr. Teemu Hirsimäki.

abilities estimated by the methods do not necessarily sum to 1 at each topic – they do in PLSA, but not either in the other methods or in the true data model – but we scale them accordingly, to be able to compare the MSE's.

In Figure 1 we see that at smaller θ, the Lift algorithm estimates the $\mathbf{Q}$ and thus the structure of the data very nicely. When θ grows very large, the data model is more difficult to estimate. The behaviors of NMF and PLSA[3] do not depend on θ, which is natural: the methods are not primarily aimed for such θ-bounded data but instead are able to estimate the structure also when the topics are totally overlapping. The K-means algorithm estimates the structure of the data poorly for all θ.

4.2 Real Data

We performed experiments on bibliographical data on computer science available on the WWW[4]. We first tested the model's prediction that $\text{lift}(A, B) \geq 1$ for all A, B; while it does not hold perfectly because there are negative correlations between words, the vast majority of these negative correlations are statistically insignificant (details omitted). We preprocessed the data by removing a small set of stop words and all numbers, and then selected the 100 most frequent terms for further analysis.

We computed the lift statistics between all term pairs and used hierarchical average linkage clustering based on the inverses of lifts. Table 1 shows how the terms are clustered into topics. The number of clusters (21) was chosen based on the distance between clusters being merged in the process of hierarchical clustering: until these 21 clusters, the intercluster distances were quite small but distances between the final 21 clusters were large. The structure in Table 1 is immediately familiar to a theoretical computer scientist: the topics concentrate on different fields of the science.

We also performed topic finding on yeast gene expression data, using the same gene expression dataset as in [24] that combines the results of several different gene expression studies. The combined dataset measures the expression level of over six thousand genes in almost a hundred experiments; thus, we used the experiments as "attributes" and the genes as "measurements". The levels were discretized so that the top 5% expressed genes in each experiment were given the value 1. The results are not shown due to space constraints, but as a brief example, the discovered topics were seen to reflect cyclical behavior of the genes in the time-series experiments.

5 Concluding Remarks

We studied a simple generative topic model and showed that the lift statistics of attributes can be described in matrix form. Based on this, we obtained a simple

[3] No simulated annealing was used in the EM algorithm of the PLSA.

[4] `http://liinwww.ira.uka.de/bibliography/Theory/Seiferas/`

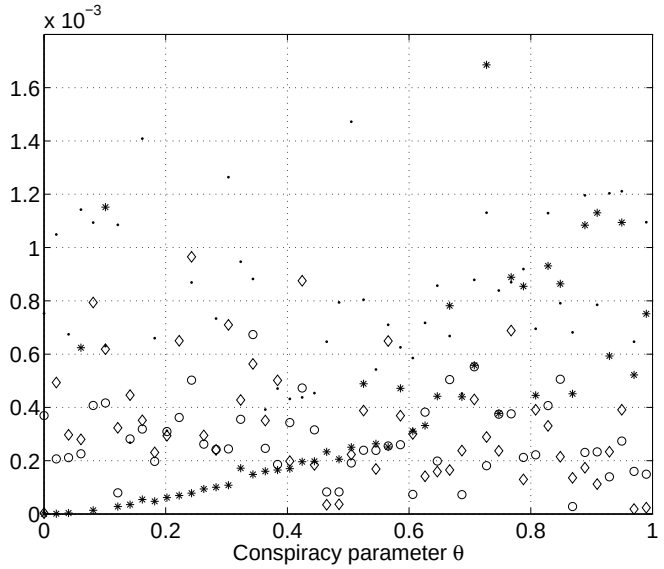

Fig. 1. Mean squared errors of $\mathbf{Q}$ at different conspiracy parameters θ. Lift $*$, NMF $\Diamond$, PLSA $\circ$, K-means $\cdot$.

topic	terms
1	algorithms approximation damath problems scheduling some tree two
2	analysis distributed libtr probabilistic systems
3	bounds communication complexity focs lower
4	algorithm efficient fast ipl matching problem set simple
5	design ieeetc network networks optimal parallel routing sorting
6	note tcs
7	finding graphs minimum planar polynomial sets sicomp time
8	graph number properties random tr
9	from information learning lncs theory
10	approach jacm linear new programming system
11	actainf binary search trees
12	abstract computation extended model stoc
13	automata finite languages mfcs
14	data dynamic infctrl logic programs structures using
15	applications icalp theorem
16	cacm computer computing science
17	crypto functions
18	jcss machines
19	algebraic beatcs computational geometry
20	de stacs van
21	codes dmath

Table 1. Terms in different topics. (The order of the topics is not relevant).

algorithm for finding topics in 0–1 data. We also showed that a problem related to the identification of topics is NP-hard, and gave experimental results.

Several open problems remain. Our model is simple, and seems to yield good results; still, more complex models might do a better job at identifying, e.g., topics containing partly exclusive attributes. The identifiability of the model is another interesting issue: could one prove something about it? Further experimental studies are also needed.

References

1. Lee, D.D., Seung, H.S.: Learning the parts of objects by non-negative matrix factorization. Nature **401** (1999) 788–791
2. Deerwester, S.C., Dumais, S.T., Landauer, T.K., Furnas, G.W., Harshman, R.A.: Indexing by latent semantic analysis. Journal of the American Society of Information Science **41** (1990) 391–407
3. Papadimitriou, C.H., Raghavan, P., Tamaki, H., Vempala, S.: Latent semantic indexing: A probabilistic analysis. In: PODS '98. (1998) 159–168
4. Hofmann, T.: Probabilistic latent semantic indexing. In: SIGIR '99, Berkeley, CA (1999) 50–57
5. Carreira-Perpiñán, M.A., Renals, S.: Practical identifiability of finite mixtures of multivariate Bernoulli distributions. Neural Computation **12** (2000) 141–152
6. Gyllenberg, M., Koski, T., Reilink, E., Verlaan, M.: Non-uniqueness in probabilistic numerical identification of bacteria. J. Appl. Prob. **31** (1994) 542–548
7. Cadez, I.V., Smyth, P., Mannila, H.: Probabilistic modeling of transaction data with applications to profiling, visualization, and prediction. In Provost, F., Srikant, R., eds.: Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Fransisco, CA (2001) 37–46
8. Hyvärinen, A., Karhunen, J., Oja, E.: Independent Component Analysis. John Wiley & Sons (2001)
9. Clifton, C., Cooley, R.: TopCat: Data mining for topic identification in a text corpus. In: Principles of Data Mining and Knowledge Discovery. (1999) 174–183
10. Dhillon, I.S.: Co-clustering documents and words using bipartite spectral graph partitioning. In: Knowledge Discovery and Data Mining. (2001) 269–274
11. Bingham, E., Mannila, H., Seppänen, J.K.: Topics in 0-1 data. In Hand, D., Keim, D., Ng, R., eds.: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD–2002), Edmonton, Alberta, Canada (2002) 450–455
12. Castelo, R., Feelders, A., Siebes, A.: MAMBO: Discovering association rules based on conditional independencies. LNCS **2189** (2001) 289–298
13. Lee, D.D., Seung, H.S.: Algorithms for non-negative matrix factorization. In: Advances in Neural Information Processing Systems. (2000)
14. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet Allocation. In: Neural Information Processing Systems 14. (2001)
15. Minka, T., Lafferty, J.: Expectation-propagation for the generative aspect model. In: Proceedings of the 18th Conference on Uncertainty in Artificial Intelligence, Edmonton, Canada (2002)
16. Buntine, W.: Variational extensions to EM and multinomial PCA. In Elomaa, T., Mannila, H., Toivonen, H., eds.: Machine Learning: ECML 2002. Number LNAI 2430 in Lecture Notes in Artificial Intelligence. Springer-Verlag (2002) 23–34

17. Comon, P.: Independent component analysis — a new concept? Signal Processing **36** (1994) 287–314
18. Jutten, C., Herault, J.: Blind separation of sources, part I: An adaptive algorithm based on neuromimetic architecture. Signal Processing **24** (1991) 1–10
19. Das, G., Mannila, H., Ronkainen, P.: Similarity of attributes by external probes. In: Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining. (1998) 23–29
20. Pajunen, P., Karhunen, J.: A maximum likelihood approach to nonlinear blind source separation. In: Proc. Int. Conf. Artif. Neural Networks. (1997) 541–546
21. Girolami, M.: A generative model for sparse discrete bianry data with non-uniform categorical priors. In: Proc. European Symposium on Artificial Neural Networks, Bruges, Belgium (2000) 1–6
22. Silverstein, C., Brin, S., Motwani, R.: Beyond market baskets: Generalizing association rules to dependence rules. Data Mining and Knowledge Discovery **2** (1998) 39–68
23. Tan, P.N., Kumar, V.: Interestingness measures for association patterns: A perspective. Technical Report TR00-036, University of Minnesota (2000) (KDD 2000 Workshop on Postprocessing in Machine Learning and Data Mining).
24. Mannila, H., Patrikainen, A., Seppänen, J.K., Kere, J.: Long-range control of expression in yeast. Bioinformatics **18** (2002) 482–483

Appendix

Proof of Theorem 1. That the problem is in NP is simple to see: the certificate is the topic vector $\boldsymbol{t}$, and the formula for $P(\boldsymbol{x} \mid \boldsymbol{t}, \mathcal{T})$ involves multiplying n numbers, each computable in $O(k)$ time.

To show NP-hardness, we reduce SAT to a topic assignment problem. Given a SAT instance of m clauses over n variables, we define a topic model with $2n$ topics and $n + m$ attributes. For each variable V_i, we create two topics T_i and T_i', and one attribute A_i. For each clause C_j, we create one attribute B_j. Each topic has probability 0.5, and each attribute has 0/1 within-topic probabilities as follows: attribute A_i has probability 1 in topics T_i and T_i' and probability 0 in other topics; attribute B_j has probability 1 in the topics T_i such that V_i appears positively in clause C_j and in the topics T_i' such that V_i appears negatively in clause C_j, and probability 0 in all other topics. We consider a data vector where all attributes have value 1.

Now, if the SAT problem has a satisfying truth assignment, it corresponds to a solution of the topic assignment problem where T_i is active if V_i is true and T_i' is active if V_i is false. This solution has likelihood 0.5^n, since exactly n topics are active, and the active topics explain all attributes A_i and B_j. Conversely, if a solution to the topic assignment problem exists such that the likelihood is at least 0.5^n, it must have at most n active topics. To explain attribute A_i, either T_i or T_i' must be active; thus the number of active topics is exactly n, and the solution corresponds to a truth assignment. Since the solution must also explain each attribute B_j, the truth assignment must satisfy the original problem. In summary, the SAT instance has a solution if and only if the topic assignment problem has a solution with likelihood at least 0.5^n. □