Available online at www.sciencedirect.com
ScienceDirect
Neurocomputing 71 (2008) 2291–2308
NEUROCOMPUTING
www.elsevier.com/locate/neucom

Factorisation and denoising of 0–1 data: A variational approach

Ata Kabán[a,*], Ella Bingham[b]

[a]School of Computer Science, The University of Birmingham, Birmingham B15 2TT, UK
[b]Helsinki Institute for Information Technology, Basic Research Unit, University of Helsinki, P.O. Box 68, Finland

Available online 25 February 2008

Abstract

Presence–absence (0–1) observations are special in that often the absence of evidence is not evidence of absence. Here we develop an independent factor model, which has the unique capability to isolate the former as an independent discrete binary noise factor. This representation then forms the basis of inferring missed presences by means of denoising. This is achieved in a probabilistic formalism, employing independent beta latent source densities and a Bernoulli data likelihood model. Variational approximations are employed to make the inferences tractable. We relate our model to existing models of 0–1 data, demonstrating its advantages for the problem considered, and we present applications in several problem domains, including social network analysis and DNA fingerprint analysis.

Keywords: Factor models; Data denoising; 0–1 data

1. Introduction

Binary data repositories arise from areas as diverse as social sciences, bioinformatics, or forensics research. The processing of binary data requires appropriate tools and methods for tasks such as exploratory analysis, feature construction and denoising. These necessarily must follow the specific distributional characteristics of the data and cannot be accomplished with tools that exist for continuous-valued data analysis.

In particular, in binary data, a '1' encodes the presence, whereas a '0' the absence of an evidence. It is common sense, however, that more often than not, the absence of evidence is not evidence of absence [23]. For example, the pixels of corrupted black and white images, the usage of words in natural language, the presence–absence patterns of social relationships or the entries of a matrix of detections of any kind all typically share this characteristic. In other binary data sets in turn, the absence of evidence is also an evidence of absence—e.g. in clean b&w raster images, the pixels that are present and those that are absent on the image, together define the content of the image. (i) How can we find out whether a given 0–1 data set has such anomalies? (ii) How can we restore a likely 'original'? Currently there is no automated method available to answer these questions, and this is what we tackle in this paper.

We regard (i) as a source separation problem: Besides content-bearing independent factors, we also need to isolate an independent factor that represents absence of evidence but not evidence of absence. If successful, this representation forms a basis for approaching the second part of the problem, (ii), which is essentially a data denoising problem. Note, the order is important here, since the existence of noise is not easily detectable, as the noisy observations are still discrete binary.

Previous successes of factor models and in particular independent component analysis (ICA) [12] make it an important statistical principle worthy of investigation for tackling both explanatory analysis and denoising problems. However, the ICA literature has been developed for continuous-valued observation signals by large, and the particular questions outlined above have never been addressed in the context of 0–1 data. Work on ICA methods for *binary observations* has been very scarce [11,6] despite their wide potential applicability, and related

*Corresponding author.
E-mail addresses: A.Kaban@cs.bham.ac.uk, axk@cs.bham.ac.uk (A. Kabán), ella@iki.fi (E. Bingham).

0925-2312/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2007.07.038

methods for discrete data in general and binary data in particular are mostly developed outside the 'mainstream' ICA community [25].

Several authors have considered the case of binary sources in the ICA literature, most recently e.g. [8,19] who give algorithms for the under-determined case of less sensors than sources. There are two major differences from this setting though, which make these methods inappropriate for the problem we consider here: First, the unknown components are binary but the noisy observations are real-valued due to the Gaussian noise assumed. As the authors point out, it is then an easy matter to determine whether there is noise or not in the data. By contrary, our observations are always binary. Hence our algorithm needs to be successful in separating out the noise component in order to reveal its presence. This is exactly the problem that we tackle. The noise component is obviously non-Gaussian, still, we will see from the presented applications that it is a very frequently occurring type of noise in real-world 0–1 data. Yet, it was never explicitly noticed in the 0–1 data analysis literature. Secondly, our setting is not under-determined but over-determined. The number of sensors in our case corresponds to the number of samples collected (e.g. number of images, number of text documents, number of nodes in a graph etc.). Although the sample size may be small, it is assumed that the number of components is smaller. In addition, contrary to methods that seek discrete binary sources, in this work, the sources will be allowed to take continuous values in the interval [0,1]. That is, rather than black & white, we will seek a grey-scale representation.

In the sequel, we formalise the problem by formulating a specific form of ICA model for multivariate binary observations. An early version appears in [15]. We employ a probabilistic framework and make use of the variational methodology to make the inference tractable. Numerical experiments will demonstrate the working of our approach and its advantages over other models of 0–1 data, for the problems considered. Application examples demonstrate the use and the added value of our approach in application areas where ICA methods have not been previously applied/applicable, such as graph or network analysis and DNA fingerprint analysis. A MatLab implementation is available from http://www.cs.bham.ac.uk/~axk/bBICA.m.

1.1. An ndependent factor model with beta sources for binary data

Consider an independent factor model for multivariate i.i.d. binary data $\boldsymbol{x}_n, n = 1,\ldots,N$, where N is the number of observations. A general form of the probability of a datum vector $\boldsymbol{x}_n$ under an independent factor model, in probabilistic terms, is the following:

$$P(\boldsymbol{x}_n) = \int P(\boldsymbol{x}_n|\boldsymbol{b}) \prod_{k=1}^{K} p(b_k)\, \mathrm{d}b_k. \tag{1}$$

Here $b_k, k = 1,\ldots,K$ represent hidden 'source' (component or factor) variables that are assumed to be independent *a priori*, and $\boldsymbol{b} = (b_1,\ldots,b_K)^{\mathrm{T}}$.

The observations are multivariate binary vectors $\boldsymbol{x}_n = (x_{1n},\ldots,x_{tn},\ldots,x_{Tn})^{\mathrm{T}}$ with T samples and N will denote the number of observation (features), $n = 1,\ldots,N$. It is well known from statistics (see e.g. [22]) that the modelling of binary observations requires a distribution that is zero outside the set of the two distinct possible values. Hence, e.g. a Gaussian likelihood model (as employed in most of the previous ICA methods) would not be appropriate in this case and for this reason we employ a conditionally independent Bernoulli likelihood model. This is parameterised by a mean vector that takes the form of a mixture of K components: $\sum_{k=1}^{K} a_{tk} b_{kn}$,

$$P(\boldsymbol{x}_n|\boldsymbol{b}_n) = \prod_{t=1}^{T} \left(\sum_{k=1}^{K} a_{tk} b_{kn}\right)^{x_{tn}} \left(1 - \sum_{k=1}^{K} a_{tk} b_{kn}\right)^{1-x_{tn}}. \tag{2}$$

The conditional independence is a standard assumption in latent variable modelling, meant to force the data dependences to be represented in the latent space. The parameters a_{tk} in (2) are the mixing coefficients of the factor model, and the mixture $\sum_k a_{tk} b_{kn}$ represents the mean parameter of the Bernoulli likelihood.[1] More intuitively, the data $\boldsymbol{x}_n$ is approximated by the combination of factors $\sum_k a_{tk} b_{kn}$, which is indeed the familiar modelling assumption of linear factor models. In both (1) and (2), the conditioning on the parameters a_{tk} is implicit.

The bulk of the design of any factor model, is the specification of the source prior distributions. These determine the statistical characteristics of the sources that we aim to infer. Here we employ independent beta latent prior densities [4]:

$$p(b_k) = B(b_k|\alpha_k^0, \beta_k^0) = \frac{\Gamma(\alpha_k^0 + \beta_k^0)}{\Gamma(\alpha_k^0)\Gamma(\beta_k^0)} (1 - b_k)^{\beta_k^0 - 1} b_k^{\alpha_k^0 - 1}, \tag{3}$$

where α_k^0 and β_k^0 are strictly positive hyperparameters. In the experiments reported, we have set both α_k^0 and β_k^0 to $\frac{1}{2}$, which is the uninformative prior.

The domain of definition of the beta density is $b_k \in [0,1], \forall k$, which is desirable for our purposes, since we may be able to *interpret* the inferred factors as grey-scale representations of the binary data. Interpretability of the components is one of the most important and desirable aspects of independent factor models in general, and this is also what we aim to achieve and exploit in this work. In addition, the particularly flexible shape (see Fig. 1) of the beta density is advantageous for the required density modelling.

The mixing process that we will assume is a convex-linear one, so that the mixing coefficients are all non-negative and satisfy $\sum_k a_{tk} = 1$, for all data-features

[1] For the ease of notations, indices (e.g. in sums or products) are always denoted by small characters and their upper limits by the associated capital letter. Unless indicated otherwise, indices run from 1 to their upper limit.

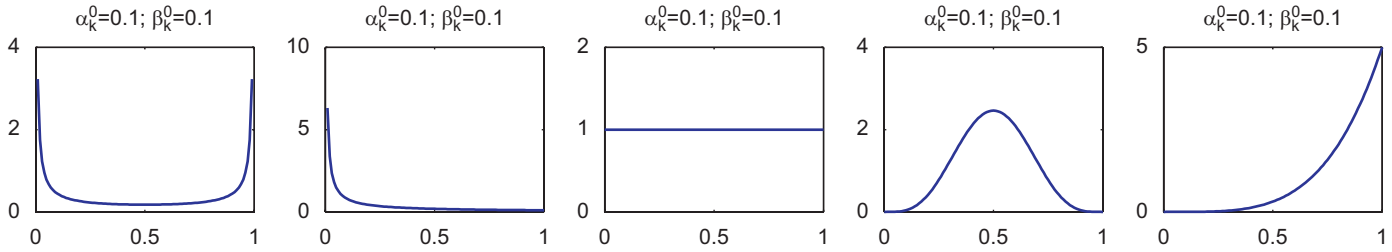


Fig. 1. The beta density with various parameters.

$t = 1 : T$. This choice is not arbitrary, since then it follows that $\sum_k a_{tk} b_{kn} \in [0,1]$, as a convex combination of $b_k \in [0,1]$. Therefore we do not need any further nonlinear transformation to obtain the mean parameter of the Bernoulli likelihood. This is an essential difference from both logistic models [27,26] and other so-called multiple-cause models for binary data [24]. While nonlinear models are also of interest and we will employ them in our comparisons, by the above model design we seek the two possible observation events, 0-s and 1-s, to be interchangeable within the model, and this would not be possible if a nonlinearity is applied to non-negative variables.

2. Inference and estimation

In order to make the problem tractable, we will employ the well-known Jensen-inequality to lower bound the data probability, and we make use of the factorial posterior approximation [13] to simplify the computations:

$$\begin{aligned}\log P(\boldsymbol{x}_n) &= \log \int P(\boldsymbol{x}_n|\boldsymbol{b}_n) \prod_k B(b_{kn}|\alpha_k^0, \beta_k^0)\, \mathrm{d}b_{kn} \\ &\geqslant \sum_t \langle \log P(x_{tn}|\boldsymbol{b}_n)\rangle_{\prod_k q(b_{kn})} \\ &\quad + \sum_{t,k} \langle \log B(b_{kn}|\alpha_k^0, \beta_k^0) - \log q(b_{kn})\rangle_{q(b_{kn})}, \end{aligned} \tag{4}$$

where $\prod_k q(b_{kn})$ is the factorial variational posterior and $\langle . \rangle$ is the expectation operator.

Now, due to the Bernoulli likelihood, the integral in the first term is still intractable. Therefore a further lower bound is created as follows. The convexity constraint imposed on the mixing proportions comes in useful, as the log of the likelihood term can be rewritten and lower bounded:

$$\begin{aligned}\log P(x_{tn}|\boldsymbol{b}_n) &= \log\left\{ \left(\sum_k a_{tk} b_{kn}\right)^{x_{tn}} \right. \\ &\quad \left. \times \left(1 - \sum_k a_{tk} b_{kn}\right)^{1-x_{tn}} \right\} \\ &= \log\left\{ \sum_k a_{tk} b_{kn}^{x_{tn}} (1-b_{kn})^{1-x_{tn}} \right\} \\ &\geqslant \sum_k Q_{tn}(k|x_{tn}) \log \frac{a_{tk} b_{kn}^{x_{tn}} (1-b_{kn})^{1-x_{tn}}}{Q_{tn}(k|x_{tn})}. \end{aligned} \tag{5}$$

Here $Q_{tn}(k|x_{tn}) \geqslant 0$, $\sum_k Q_{tn}(k|x_{tn}) = 1$ is a discrete variational distribution with values in $\{1, \ldots, K\}$, where K denotes the number of components.

Replacing (5) into (4), the obtained lower bound is now tractable to compute and will be referred to as $\mathscr{L}^{\text{bound}}$:

$$\begin{aligned}\mathscr{L}^{\text{bound}}(\boldsymbol{x}_n) = \sum_{t,k} Q_{tn}(k|x_{tn}) \{ &\log a_{tk} + \langle \log b_{kn}^{x_{tn}} (1-b_{kn})^{1-x_{tn}} \rangle \\ &- \log Q_{tn}(k|x_{tn}) \} + \sum_k \{ \langle \log B(b_k|\alpha_k^0, \beta_k^0) \\ &- \log q(b_{kn}) \rangle \}, \end{aligned} \tag{6}$$

where $\langle . \rangle$ denotes expectation w.r.t. $q(b_{kn})$.

2.1. Variational EM solution

By maximising $\mathscr{L}^{\text{bound}}$, a generalised EM algorithm with partial E-steps [13,1] can be derived. In the variational E-step, the mixing coefficients a_{tk} are kept fixed and we compute the variational posteriors $Q_{tn}(k|x_{tn})$ and $q(b_{kn})$ in order to make the bound as tight as possible. In the M-step, we maximise $\mathscr{L}^{\text{bound}}$ as a function of mixing coefficients a_{tk}, while keeping the variational posteriors fixed. Each of these two steps is guaranteed not to decrease the bound.

2.1.1. Variational E-step

Straightforward variational optimisation (Appendix A.1) yields the optimal form for the variational posteriors. The optimal functional form for $q(b_{kn})$ turns out to be a beta density:

$$q(b_{kn}) = B(b_k|\alpha_{kn}, \beta_{kn}) \tag{7}$$

with variational parameters

$$\begin{aligned}\alpha_{kn} &= \alpha_k^0 + \sum_t x_{tn} Q_{tn}(k|x_{tn} = 1), \\ \beta_{kn} &= \beta_k^0 + \sum_t (1 - x_{tn}) Q_{tn}(k|x_{tn} = 0). \end{aligned} \tag{8}$$

Further,

$$Q_{tn}(k|x_{tn}) \propto a_{tk} (\mathrm{e}^{\langle \log b_{kn} \rangle})^{x_{tn}} (\mathrm{e}^{\langle \log(1-b_{kn}) \rangle})^{1-x_{tn}}, \tag{9}$$

where the required variational posterior expectations in (9) are evaluated as $\langle \log b_{kn} \rangle \equiv E_{q(b_{kn})}[\log b_{kn}] = \psi(\alpha_{kn}) - \psi(\alpha_{kn} + \beta_{kn})$ and $\langle \log(1 - b_{kn}) \rangle \equiv E_{q(b_{kn})}[\log(1 - b_{kn})] = \psi(\beta_{kn}) - \psi(\alpha_{kn} + \beta_{kn})$.

Note these two posterior quantities are interdependent. Therefore (8) and (9) need to be alternated in an inner loop within the variational E-step.

It is also convenient to notice that the expression of $Q_{tn}(k|x_{tn})$ may be replaced into (8) so that the somewhat burdensome multidimensional matrix (9) needs not be stored. Hence, the obtained variational E-step equivalently can be accomplished by iterating the following two updates. We typically observed convergence within 5–6 iterations or less.

$$\alpha_{kn} = \alpha_k^0 + \mathrm{e}^{\langle \log b_{kn}\rangle} \sum_t \frac{x_{tn} a_{tk}}{\sum_{k'} a_{tk'} \mathrm{e}^{\langle \log b_{k'n}\rangle}}, \tag{10}$$

$$\beta_{kn} = \beta_k^0 + \mathrm{e}^{\langle \log(1-b_{kn})\rangle} \sum_t \frac{(1-x_{tn}) a_{tk}}{\sum_{k'} a_{tk'} \mathrm{e}^{\langle \log(1-b_{k'n})\rangle}}. \tag{11}$$

2.1.2. M-step

The estimation of the mixing coefficients is now carried out. Maximising $\mathscr{L}^{\text{bound}}$ w.r.t. a_{tk} under the constraint that $\sum_k a_{tk} = 1$ (Appendix A.2) and combining with the expression of $Q_{tn}(k|x_{tn})$, yields the update equation:

$$a_{tk} \propto a_{tk} \left\{ \sum_n \frac{x_{tn}}{\sum_{k'} a_{tk'} \mathrm{e}^{\langle \log b_{k'n}\rangle}} \mathrm{e}^{\langle \log b_{kn}\rangle} + \frac{1-x_{tn}}{\sum_{k'} a_{tk'} \mathrm{e}^{\langle \log(1-b_{k'n})\rangle}} \mathrm{e}^{\langle \log(1-b_{kn})\rangle} \right\}. \tag{12}$$

The algorithm is then to iterate the variational E-step and the M-step to convergence.

2.2. Variational Bayesian solution

So far, the mixing coefficients a_{tk} have been treated as free parameters. Therefore the likelihood bound $\mathscr{L}^{\text{bound}}$ is not suitable for selecting the optimal number of components. To overcome this, we may place a prior over the mixing coefficients. Because of the convexity constraint that we imposed (see Section 2), a Dirichlet density is appropriate. The model then resembles some analogies with generative aspect models for count-based data [5,6,20], which have been quite popular recently for text document analysis and collaborative filtering, but have never been applied to either denoising problems or 0–1 data analysis. We have set the Dirichlet hyperparameter to 1 in all our experiments, in order to encourage a uniform spread of the mixing coefficients.

Since now there are no free parameters left in the model, the optimal number of components can be determined by choosing the model order that maximises the log of the data evidence bound [13] (see Appendix B.1). Alternatively, we may initialise the model with a relatively large number of components and the priors will drive the unnecessary components to extinction. From our experiments we found this latter procedure more convenient for two reasons: It does not require us to repeat the runs for several candidate number of components. Secondly, we do not have much information about the tightness of the bound and have observed the evidence bound as a criterion for model selection may occasionally underestimate the number of components in this model.

Nevertheless, the priors are necessary for performing a Bayesian model selection. The modification brought to the previously presented estimation procedure is that now a variational M-step is required. This is derived analogously to the variational E-step. Details are given in Appendix B.2. It should be mentioned that the variational Bayesian estimation methodologies have a relatively long successful history with various independent factor models over continuous-valued data [17,1], as well as a number of other latent variable models [2].

3. Analyst input and posterior data reconstruction

Perhaps the greatest reason for the popularity of ICA methods for exploratory data analysis is that the independent components are often easier to comprehend and interpret by humans separately, rather than in their mixture. This has been exploited in numerous applications, most notably in the context of medical signal denoising [14]. Once the independent signals of different genuine and artefact sources are separated from the data, artefact-corrected signals may be derived by eliminating the contributions of the artefact sources. Our methodology is conceptually similar, although the formalism differs according to our probabilistic framework.

Let us denote the posterior expectations obtained from our algorithm by $\langle a_{tk}\rangle$ and $\langle b_{kn}\rangle$, respectively: $\langle b_{kn}\rangle = E_{q(b_{kn})}[b_k] = \int \mathrm{d}b_k\, b_k B(b_k|\alpha_{kn},\beta_{kn}) = \alpha_{kn}/(\alpha_{kn}+\beta_{kn})$ and analogously $\langle a_{tk}\rangle = \gamma_{tk}/\sum_{k'}\gamma_{tk'}$—when a Dirichlet prior was employed, or otherwise we work with the estimates a_{tk}. These are themselves discrete probabilities, so that $\sum_k \langle a_{tk}\rangle = 1$. After inspecting the independent components $\langle \boldsymbol{b}_k\rangle$, the elimination of undesired components may now be accomplished by specifying a probability value, $P(u|k)$, for each component k, and using these to modify our unsupervised estimates.

Let us denote by $P_t(k)$ the posterior expectations $\langle a_{tk}\rangle$, for the Bayesian version or simply the estimates a_{tk} in the variational EM version. In both cases, Bayes rule will provide the post-processed data representation:

$$\langle a_{tk}\rangle_{\text{postproc}} := P_t(k|u) = \frac{P_t(k)P(u|k)}{\sum_{k'} P_t(k')P(u|k')}. \tag{13}$$

Typically $P(u|k) = 0$ will be specified for components that are capturing undesirable noise factors, while $P(u|k) = 1$ will specify a clearly meaningful component. It is easy to see that having a value of $P(u|k) = 0$ implies that $\langle a_{tk}\rangle_{\text{postproc}} = 0, \forall t$. This essentially means that we remove the components rated as noise and re-normalise the mixing coefficients $a_{tk'}, k' \neq k$ of the remaining ones.

Naturally, the formalism straightforwardly permits also the specification of analyst inputs at more detailed levels.

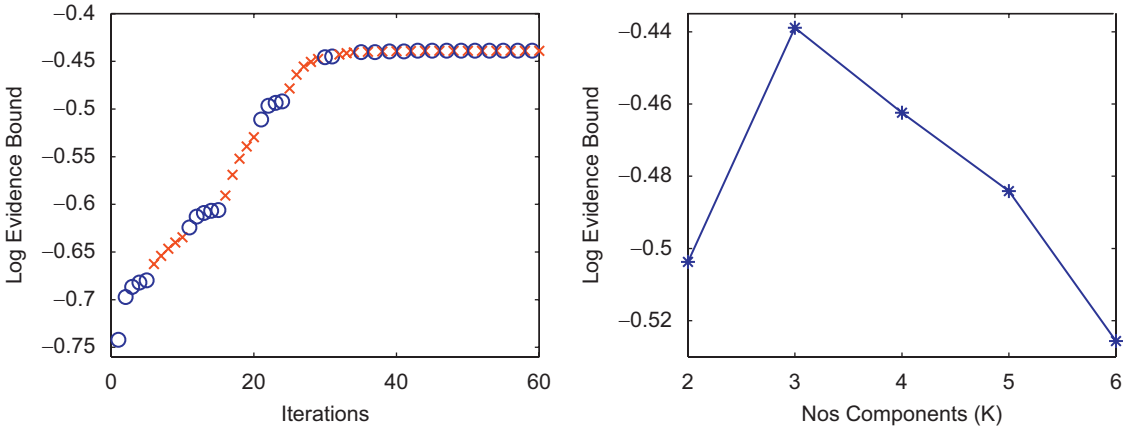


Fig. 2. Left: the monotonically increasing log evidence bound versus iterations: 'o': within variational E-steps, 'x': within variational M-steps. Right: the log evidence bound peaks at $K = 3$, which is the correct number of components in the toy data set.

E.g. nothing prevents us from specifying a separate set of probabilities, $P(u|k,t)$, for each t. However, we may typically expect human experts to feed back on the level of entire components, since those are hoped to provide some interpretable representations.

For computing the posterior data reconstruction, we re-express the post-processing described, in terms of a conditional posterior $q(\boldsymbol{a}_t|\boldsymbol{u}) := D(\boldsymbol{a}_t|\boldsymbol{\gamma}_t \circ P(u|.))$, whose expectation is exactly $\langle a_{tk}|\boldsymbol{u}\rangle = \langle a_{tk}\rangle_{\text{postproc}}$. Here, D denotes the Dirichlet distribution, $\circ$ denotes element-wise product and $\boldsymbol{u}$ is the random vector of $u|k$ when $k = 1:K$. Then the posterior probability that a data entry is reconstructed as a 1 is the following (Appendix C):

$$P(\hat{x}_{tn} = 1|\boldsymbol{X}, \boldsymbol{u}) = \sum_k \langle a_{tk}|\boldsymbol{u}\rangle \langle b_{kn}\rangle \tag{14}$$

and so the binary reconstruction is given by thresholding this value.

4. Experiments and evaluation

4.1. A toy experiment

We generated a simple toy data set from the model, with $K = 3$, of size $T = 150$ and $N = 30$. The log evidence bound is monitored against iterations till convergence, on the left-hand plot of Fig. 2. As expected, a monotonic increase can be observed. We have set the maximum number of inner loops for both the variational E and M steps to 10 and each of these inner loops is stopped earlier if the change in log evidence is less then 10^{-3}. The values are monitored with two different symbols for the variational E and M steps, respectively, so it can nicely be seen how the inner loops get shorter over time, towards convergence. On the right-hand plot of Fig. 2, we see the converged log likelihood bound for different trials of model orders in the range 2–6. The peak is at $K = 3$, and so the model order is correctly recovered. At more than three components, the extra components are automatically eliminated: Their posterior equals to their prior and the expectation of the associated mixing coefficients goes to zero.

The subsequent experiments demonstrate the working of our model, together with detailed quantitative evaluation on semi-synthetic data. It should be noted that—similarly to other 'blind' separation models and methods—the approach presented is aimed to be a 'generic' tool for analysing and denoising binary data. It is nearly certain that for any specific application area, an improved refinement could be made, e.g. by employing more domain-specific dependency structure within the priors instead of our independent beta priors. Such specific tailoring is outside the scope of this paper. Instead, the experiments that follow are meant to demonstrate that given a 0–1 data set, our method succeeds at identifying binary noise and restoring a more likely original.

4.2. Restoration of corrupted binary images

A data set of handwritten digit images[2] is employed in the subsequent experiments. The subset '0'–'4' is employed, which has 200 examples for each digit, which totals $T = 1000$ instances. The number of pixels on each image is $N = 15 \times 16 = 240$. We artificially created a corrupted version of this data set, by simulating a uniformly varying process of degradation, which turns off some of the pixels that were initially 'on'. Fifteen randomly chosen examples are shown from the initial data set, along with their corrupted version, in Fig. 3. We run the variational Bayesian version of our method for 500 outer iterations and with a maximum of inner loops set to 5. Fig. 4 then shows the ICA representation obtained[3]: Several components can clearly be recognised as typical digits, and one other, 'blank' component separates out the corruption factor. Inspecting the mixing proportions for the data instances shown earlier, it is clear that the white

[2] http://www.ics.uci.edu/mlearn/MLSummary.html.

[3] Unnecessary components, which have their posterior equal to their prior (thus look completely grey) are not shown.

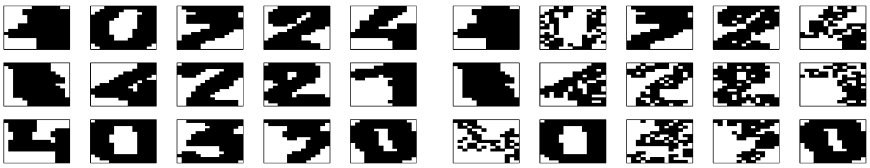


Fig. 3. Examples of clean (left) and corrupted (right) images.

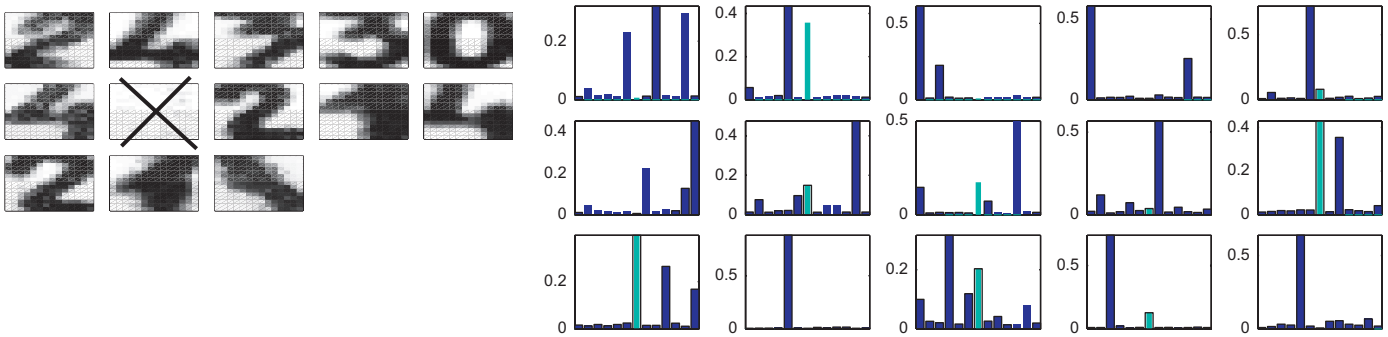


Fig. 4. Left: the components estimated from the corrupted binary image data set. Right: the mixing coefficients associated with the examples shown on the right-hand plot of Fig. 3. The mixing coefficients associated with the noise component are highlighted in light colour.

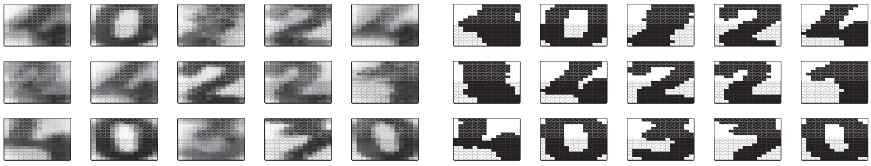


Fig. 5. Reconstructed grey-scale (left) and binary (right) images after post-processing.

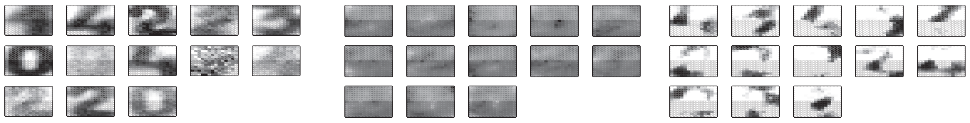


Fig. 6. The representation (factors) created by other competing binary factor models. From left to right: MB, LPCA and BNMF.

component is present in exactly those images that suffered a degradation.

To remove the noise component, we apply the procedure described earlier. The results can be followed in Fig. 5: The grey-scale posterior reconstruction of the data has indeed filtered out the degradation factor and presents a smoothed reconstruction of the initial clean data. On this plot, the grey levels correspond to probabilities of pixels being 'on'. Thresholding these probabilities at 0.5 gives us the binary reconstruction of the data shown on the right-hand plot. The degradation has now been eliminated.

A comparative set of experiments has then been conducted in order to assess the performance of our method in reconstructing the clean data from its corrupted version. We included a comprehensive set of binary data analysis methods in this comparison: mixtures of Bernoulli (MB), Bernoulli (logistic) PCA [27] (LPCA), our binary ICA with and without post-processing (BICA-postproc and BICA, respectively), and a Bernoulli version of non-negative matrix factorisation [18], that we created for the purpose of this comparison (BNMF). For the latter, a shifted and rescaled sigmoid nonlinearity was used, which transforms the non-negatively constrained factors and mixing proportions into the [0,1] interval. Fig. 6 shows the representations created by these other models. None of the methods except BICA was able to separate out the noise factor. In consequence no obvious correction post-processing is applicable to the other methods. Fig. 7 shows their grey-scale reconstruction obtained. Despite some smoothing, the corrupted images are still of low quality.

For a first quantitative assessment, we split the data into two halves: 500 corrupted images were used for training and another 500 corrupted images formed an independent test set. We will refer to denoising the training set as 'weak denoising', whereas denoising the previously unseen test set (both the training and testing sets are corrupted in this case) will be referred to as 'strong denoising'. Note that the variational Bayesian version of our algorithm, that has a

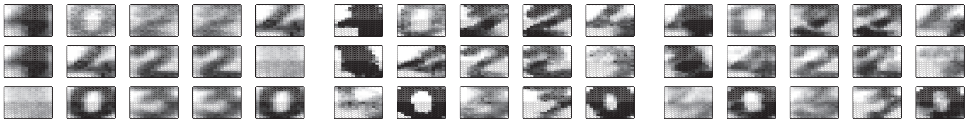


Fig. 7. The reconstructed grey-scale images by other competing binary factor models. From left to right: MB, LPCA and BNMF.

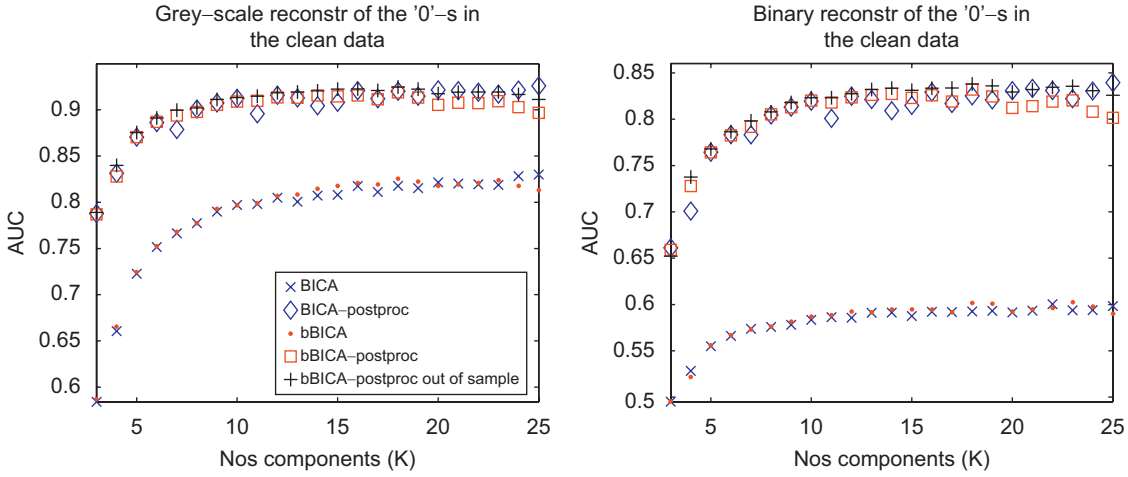


Fig. 8. The effect of post-processing on the denoising performance of BICA and bBICA.

prior postulated on a_{tk} can be used for strong denoising: For the previously unseen data instances, the variational parameters of the variational posterior $q(\boldsymbol{a}_t)$ are estimated. To differentiate between this and the variational EM version of our model, BICA will refer to our model estimated by variational EM and bBICA will stand for the variational Bayesian version.

The post-processing was performed as described earlier and to automate the process, for this particular data set, a threshold of 0.1 was employed: If the average of a component, i.e. $\sum_n \langle b_{kn} \rangle / N$ is below this threshold then the component is removed. This worked well in this experiment although of course using human expertise may potentially further improve the results, especially in cases when we do not know beforehand what average statistics the useful/noisy components might have.

Fig. 8 shows the beneficial effect of the proposed post-processing for both BICA and bBICA. For the latter, strong denoising is also demonstrated on the plot. On these plots, and throughout, the performance is measured in terms of the area under the average expected ROC curve of all instances (in sample for weak denoising, out of sample for strong denoising) (AUC)[4] [9] of the posterior reconstruction. More precisely, the posterior reconstruction of pixels in image regions where the clean image is white were merged together to produce the average expected ROC curve and the area under this ROC curve is plotted against the number of components in the range 3–25. Note the clean data are not used anywhere else, only for evaluation.

[4]For binary reconstruction $AUC = 1 - (fp + fn)/2$, where fp is the false positive rate and fn is the false negative rate.

We see the two versions of our algorithm perform similarly on weak denoising and they are accurate over a wide range of model orders. It is also notable that the strong denoising results are no inferior in this experiment.

We compare the denoising performance of our approach to other binary factor models. Figs. 9 and 10 show this comparison in terms of weak and strong denoising, respectively. As we can see, the proposed post-processing, by the removal of the automatically separated noise component, BICA becomes the most successful in this exercise—comparable with the nonlinear and time-consuming LPCA at grey-scale reconstruction and net superior at binary reconstruction. Note that LPCA scales cubically per iteration, due to a matrix inversion required at each iteration. Furthermore, finding a suitable threshold to obtain an accurate binary reconstruction would require further computations. It is also instructive to inspect the extent to which the models 'get fooled' to reconstruct the corrupted data sets. This is shown in Fig. 11. Clearly, LPCA ranks first in this, due to its flexibility, while our post-processing strategy, as we have seen, results in a poorer reconstruction of the erroneous data but instead excels in reconstructing the never seen clean data.

4.2.1. *Varying the training set size*

It is an important issue to study the variation of these results with the training set size. In our next experiment we vary the size of the training set and measure the noise removal capability of the model in corrupted digit data. Of the total of 1000 datum instances, $200, 300, \ldots, 900$ were sampled randomly for training and the rest used for testing. The number of components was chosen as $K = 10$

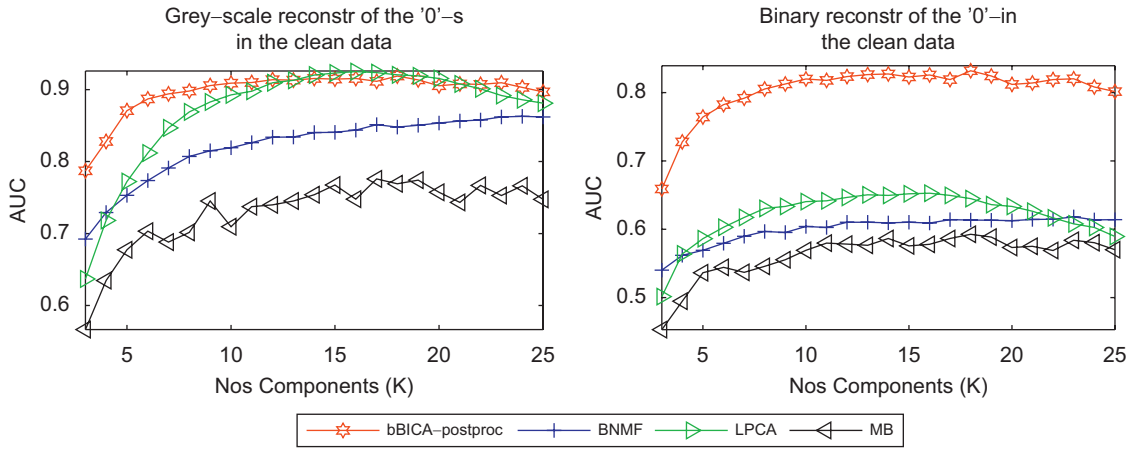


Fig. 9. Comparison of bBICA with other binary factor models on weak denoising.

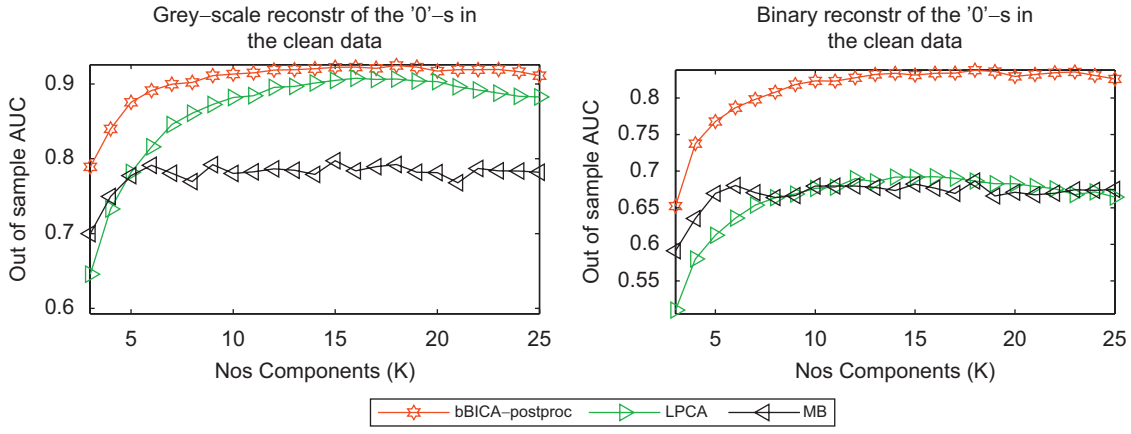


Fig. 10. Comparison of bBICA with other binary factor models on strong denoising.

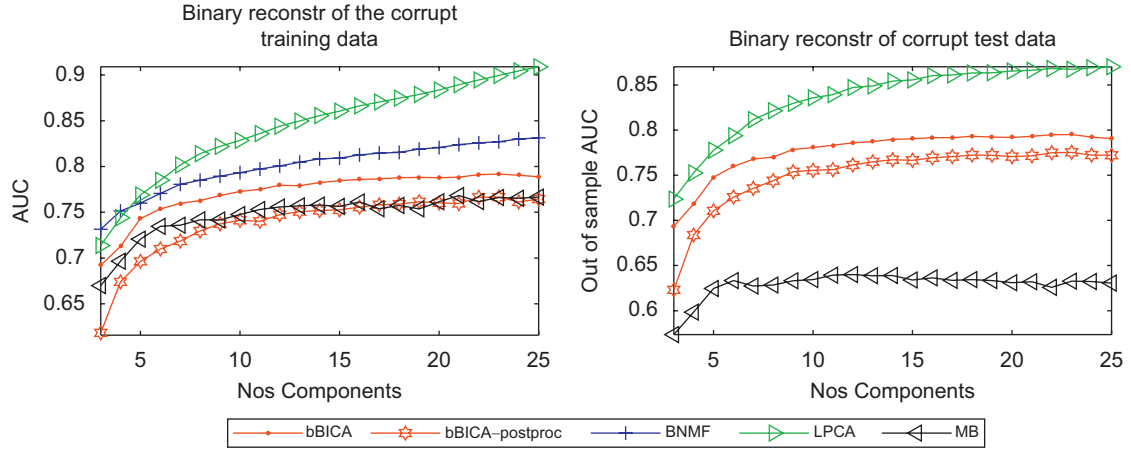


Fig. 11. Comparison of the models for 'verbatim' reconstruction of the corrupt data sets. All models get fooled to some extent. However, our post-processing strategy, reduces the accuracy of reconstructing the erroneous data in favour of reconstructing the never seen clean data.

throughout this experiment, as the earlier experiments demonstrated that the choice of K is not crucial. As before, we measure the post-processed model's ability to reconstruct the noiseless data, at the zero (white) entries of the training set (weak denoising) and test set (strong denoising), respectively. Fig. 12 shows the variation as a function of the training set size. The error bars show the mean and one standard deviation over the 10 bootstrap repeats for each training set size tested. As one would expect, we see that the weak denoising performance

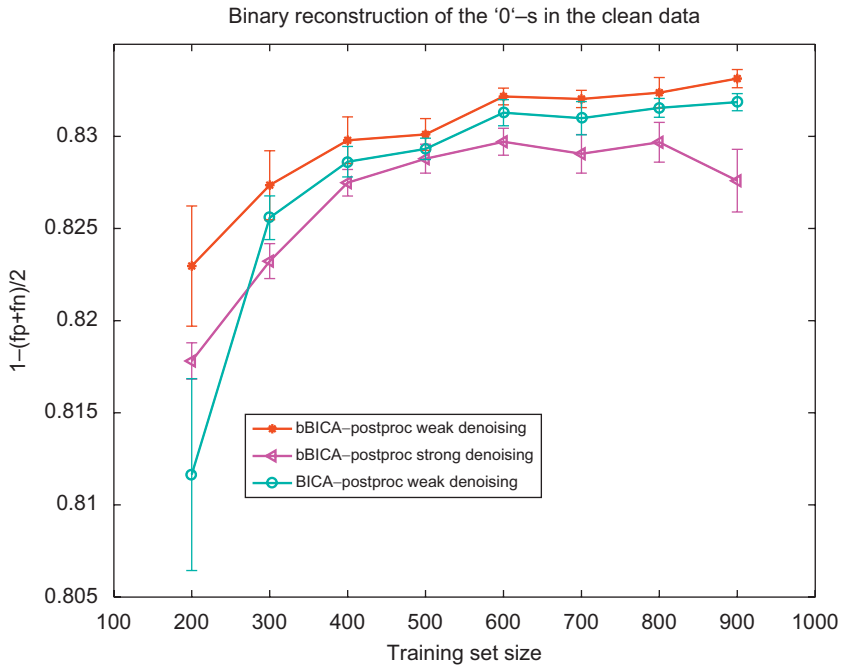


Fig. 12. Varying the training set size (horizontal axis) in noise removal of handwritten digits. Vertical axis: AUC.

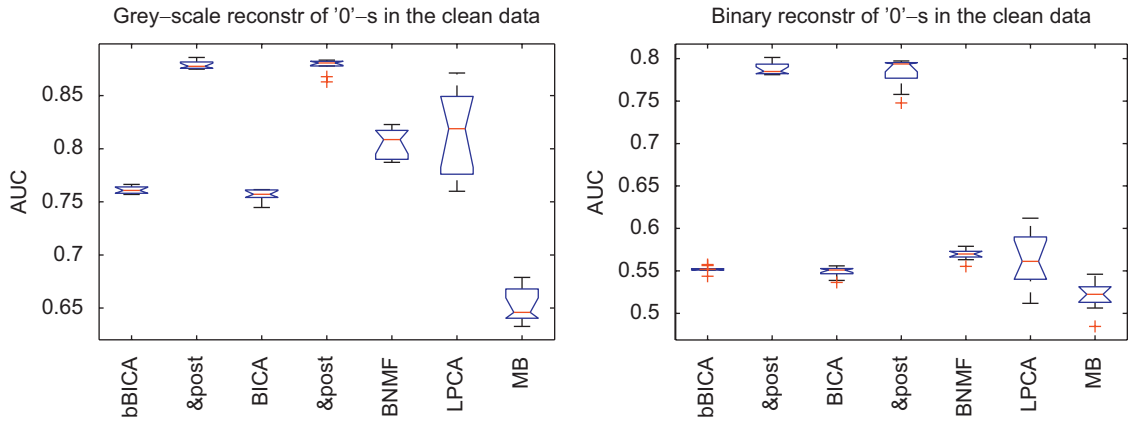


Fig. 13. Weak denoising on small size data set ($N = 125$).

of both bBICA and BICA improves at larger training set sizes and the performance of bBICA at test data (strong denoising) levels up after a training set size of cca. 500 instances.

One may then wonder what happens when the training set gets small, where one may expect to encounter a larger variation. On the other hand, in the case of a small training set, the form of the model and the priors matter much more—a more appropriate model (in terms of the purpose of the modelling) can be expected to have a greater advantage. Furthermore, it is interesting to see the variation comparatively with the other models and to assess the statistical significance of the differences between methods.

To provide an insight into this issue, Figs. 13 and 14 show the variation of weak and strong denoising, respectively, when both the training and the test set size is as small as $N = 125$. For this experiment, both sets were sampled randomly from the previously employed larger $(500 + 500)$ sets (training and testing sets are disjoint, of course) and repeats were performed in the range $K = 9$–19. All these results were then collected together for each model in turn and these distributions are shown on the plots of Figs. 13 and 14. Apparently, the advantage of our approach is more pronounced in this setting. A pairwise application of the Kolmogorov–Smirnov test has indicated the difference in performance of post-processed BICA and bBICA and the performance of all other models is significant for both weak and strong denoising (the p-value has been of the order 10^{-9}). No significant difference was detected between BICA-post and bBICA-post on weak denoising.

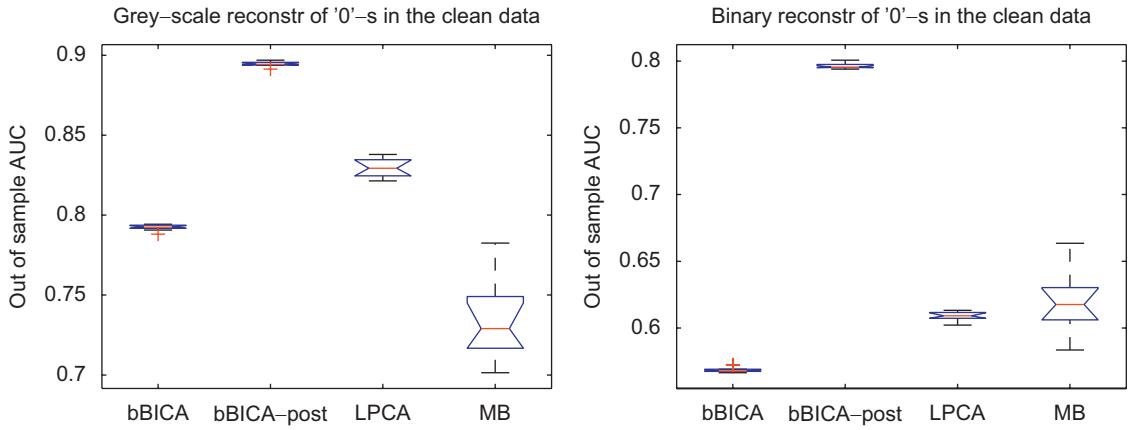


Fig. 14. Strong denoising on small size data set ($N = 125$).

5. Applications

In this section we present examples of real-world application areas where our method may be used, and its added value over existing alternative 0–1 data analysis models and methods. In principle, it is applicable whenever the data under study consists of multivariate 0–1 vectors.

5.1. Identification of treatment groups from DNA fingerprints

Microbial community fingerprints are intensely studied in agriculture, e.g. in the context of optimising the productivity of the soil. They can be represented as binary vectors [28], where the observations are presence/absence indicators of microbial populations across a number of samples is different treatment groups. We use data from [28], where the objective was to investigate the impact of different agronomic treatments on the microbial community structure of corn in rizosphere. This consists of $T = 89$ samples from four different treatment groups, over $N = 84$ microbial populations. See [28] for details. In [28], four different feature selection and classification combinations were devised and applied to this data. These are supervised methods that use class label information at the training stage. Their best results, in terms of the number of correct classifications in leave-one-out testing, are listed in Table 1, along with the true number of instances in each class—10 instances are misclassified in total. These form an objective basis for evaluating our bBICA analysis, as follows. Now we apply bBICA to the same data set, without using any feature selection or any other pre-processing, and without making use of class information. In order to avoid possible spurious local optima, we repeated our algorithm 50 times, selecting the best local optimum of the data evidence. The obtained factorisation is shown in Fig. 15. Differently from some of the other examples given, no noise component is detected in this data, meaning that the absence of evidence of any of the microbial populations most probably represents an evidence of its absence. Moreover, it is quite apparent from the figure that the mixing coefficients of bBICA discover four distinct classes. After an appropriate permutation of the components (by computing the confusion matrix), we find a remarkable correspondence between the strongest component and the true class labels, and the number and percentage of correct matches is given in Table 1. Note, the total number of mismatches is 6, which is lower than that previously found with the best supervised method. The results of a Bernoulli mixture clustering (selected based on highest likelihood from 50 repeats to avoid local optima) is also shown as a baseline in the table. The Bernoulli mixture confuses the classes 3 and 4 and displays a rather poor match with the true treatment groups. In addition, the bBICA components represent the characteristic presence–absence patterns of microbial populations associated with the four discovered treatment groups, and thus naturally reveal information about the impact of the various treatments.

Table 1
Clustering and classification results on DNA fingerprints, in terms of the number of correct matches with the true class labels and the total matches given in percentage. The leave-one-out (L-O-O) classification results are taken from [28] and represent the best results they obtained on this data. Despite our bBICA is an unsupervised method, the agreement with the true class labels is higher

	Class 1	Class 2	Class 3	Class 4	Total	Accuracy (%)
True	23	22	22	22	89	–
L-O-O classif. [28]	22	18	22	17	79	83
bBICA	23	19	22	19	83	**93**
Bernoulli mixtures	23	17	22	0	62	69

As described above, this result was obtained by selecting, from multiple random restarts, the run that obtained the highest evidence bound (i.e. the best local optimum). To see how well this unsupervised, model-based criterion works, it is also interesting to inspect the correlation between the converged evidence bound values and the clustering accuracy. Fig. 16 (left-hand plot) shows the scatter-plot from 20 repeats. We observe the existence of spurious local

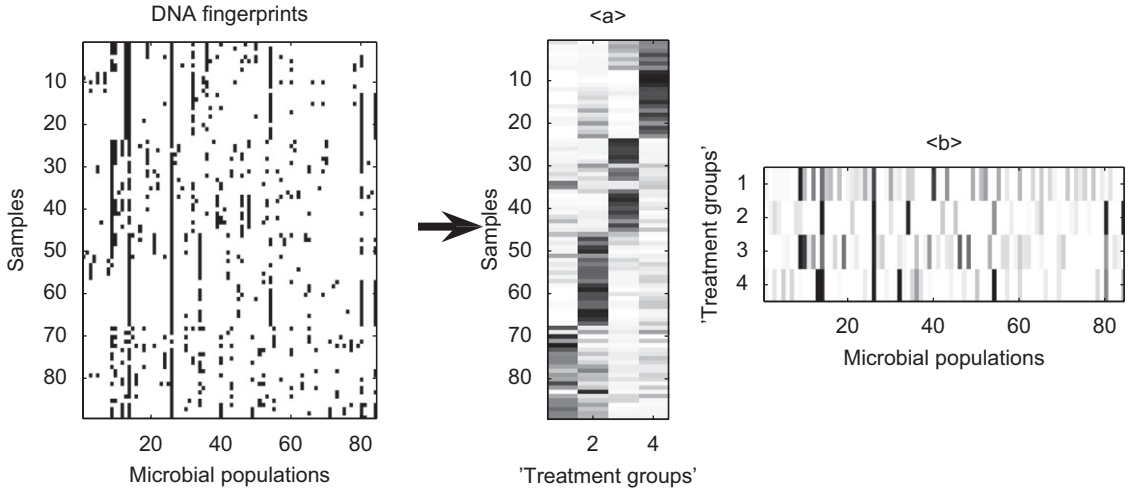


Fig. 15. The bBICA decomposition of DNA fingerprints discovers the four treatment groups.

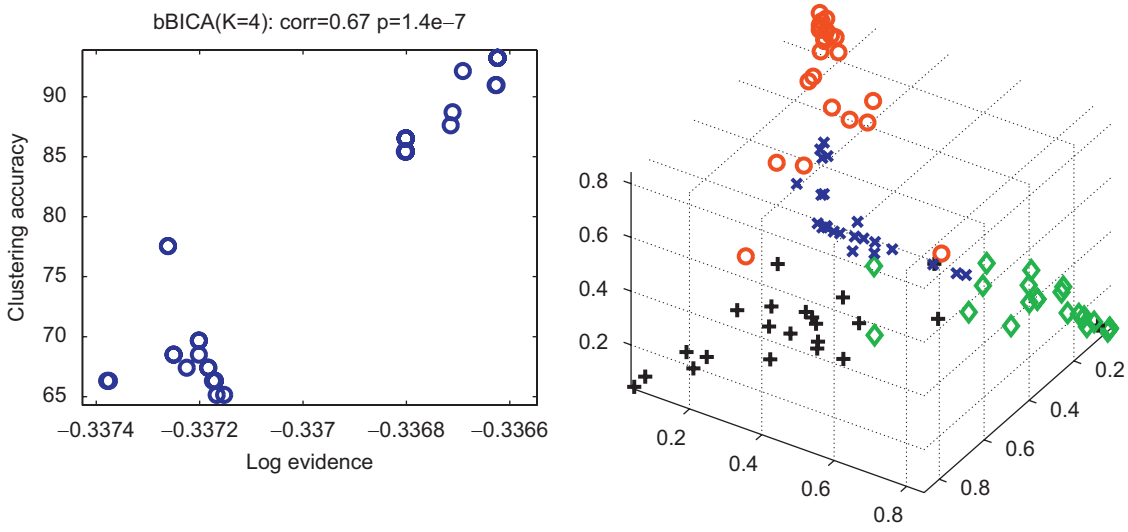


Fig. 16. Left: correlation between the data evidence bound and the clustering accuracy in the DNA fingerprint experiments. Right: 3D visual plot of the posterior means of mixing coefficients.

optima at some distance from the better ones. But more importantly, we find a significant positive correlation between these two quantities—the evidence bound and the cluster accuracy—in terms of a good match with the true class labels. Hence, by selecting the model that achieves a better optimum in terms of the evidence bound is also likely to be a model that produces a better match with the true class labels—which means the model is well suited for describing this data. The right-hand plot displays the posterior mean mixing coefficients corresponding to the result with highest evidence (seen also in Fig. 15) as a 3D plot. We see the misclassified points are actually not that far from their correct class.

Finally, we also tested the model for initial values of K other than the true number of clusters. When K was initialised to a larger number (e.g. we tried $K = 12$ and 16), the model tended to settle at a final number of components larger than 4. To somehow quantify these results, we then assigned each component the class of the majority of its data, using the posterior estimates of the mixing proportions. This evaluation strategy was previously used in [7] to match up the clusters identified by a Bayesian model with a smaller number of true classes. It turns out, as shown in Fig. 17 that the components tend to subdivide the true treatment groups, without confusing them.

5.2. *Finding groups and identifying opportunities from social networks*

Graphs or network models are widely used to represent relations between interacting entities—e.g. epidemic networks, computer networks, gene networks and social

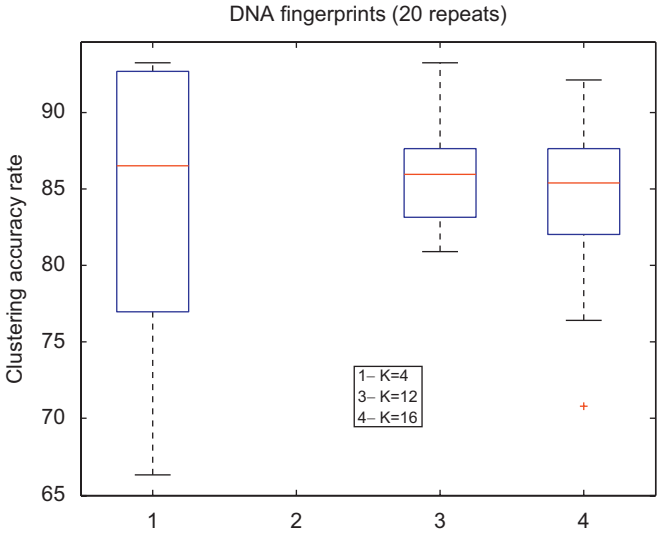


Fig. 17. Distributions of the cluster accuracy results when the initial number of components is varied. The best results with different initial K are comparable.

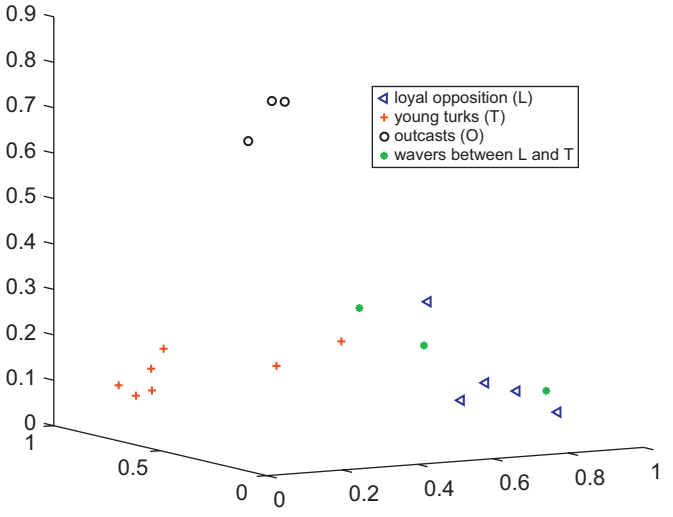


Fig. 18. Visual display of the posterior expectations of the mixing coefficients estimated from the Monks network data set.

networks, to name just a few. In a social network, each node represents a person or a social group and each link or edge has information about a relationship. Here we will focus on 0–1 relations, i.e. two nodes are either connected or they are not. We consider the edges are directed, that is, if a node has a link to another node, the converse is not implied. There has been a lot of interest in modelling and analysing network data in general and social networks in particular—see e.g. [10,16,25] for some recent studies. However, we know of no applications of ICA approaches to this problem.

For a first example, we took the monks network used in [10], which received much attention in the social networks literature. It describes the social relations between 18 monks in an isolated American monastery (see [10] and references therein for details on the data and its previous uses). There are three main groups: the young turks (T) (seven members), the loyal opposition (L) (five members) and the outcasts (O) (three members). In addition, three monks wavered between L and T. We run bBICA, in 50 randomly initialised repeats to avoid local optima and selected the run with best log evidence. In Fig. 18 we show the posterior expectations of the inferred mixing proportions—since $K = 3$ in this case, these can be easily visualised. The markers reflect the true labels for the convenience of visual evaluation. We see the three groups are well separated and there is a good agreement with the true structure of the data. Two out of the three wavers are indeed situated between the groups of L and T. It should be stressed, this latter property is not exhibited by clustering methods, which, by contrary, tend to divide the data into disjoint groups. E.g. the clustering result in [3], for the same data, has grouped two of the wavers with the group of 'L'-s and one other with the group of 'O'-s. Therefore, our bBICA model is more than a clustering method; it can preserve some of the topology of the network nodes.

After this illustrative example on the rather small and clean, previously well-studied monks network, in the sequel, we analyse a real-world social network, collected from an Internet Relay chat room. Initially, the data are a temporal sequence of 25,355 contributions made by 844 participants. For a topic-independent analysis of the social relations, the sequence of ID-s may be analysed [16]. Contrarily to this previous study, here we represent the ID sequence as a binary graph, in order to infer the underlying components of the presences and absences of relationships (rather than their 'strengths'). Since this is a real-world example, we anticipate that apart from components that correspond to clear groups or communities, there will be noise components as well. As we will see, the noise components of this binary representation are very useful to identify and can be interpreted in this context as missed relationship-opportunities. The removal of such components will reveal links that are invisible when the noise component is present.

The nodes of our chat network are the 844 unique ID-s. We construct the binary graph in two ways. A first order graph will have a '1' whenever a consecutive contribution of a pair of participants exists in the sequence. This is a very crude representation, since the intended order of contributions may interleave in practice and random temporal delays may be present. Therefore, our second (and more realistic) approach is to pre-process the sequence using the mixed transition Markov model of [3], i.e. to infer the intended connections by taking into account transitions situated at deeper temporal lags. The maximum lag was set to 8, which should be sufficiently long to recover delays that are due to differences in typing speed or those due to network bandwidth limitations. At each time

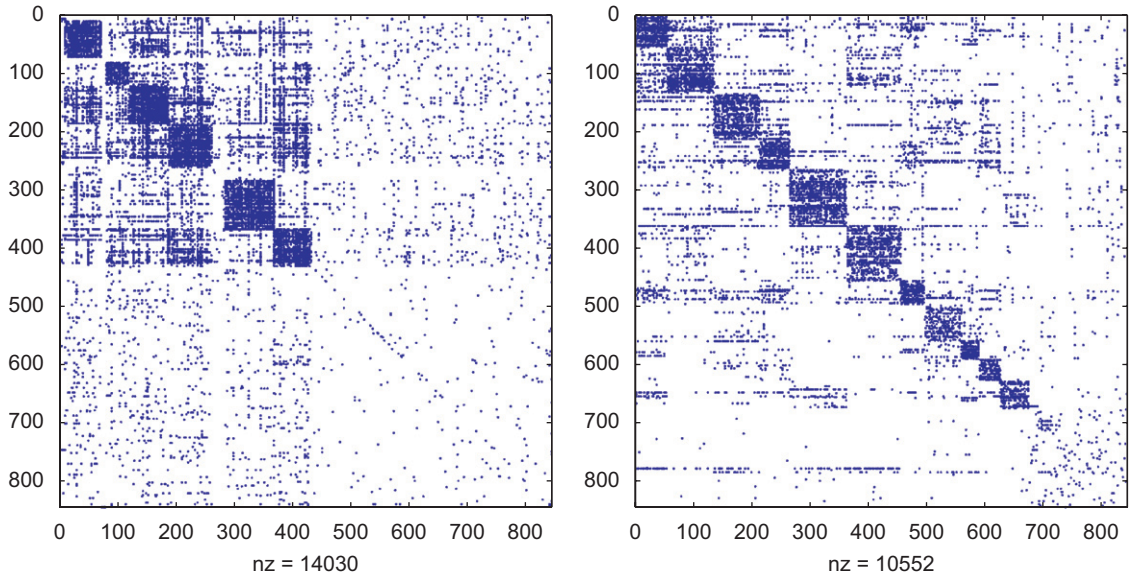


Fig. 19. Left: the reordered first-order network. Right: the reordered network when mixed-transition pre-processing was used.

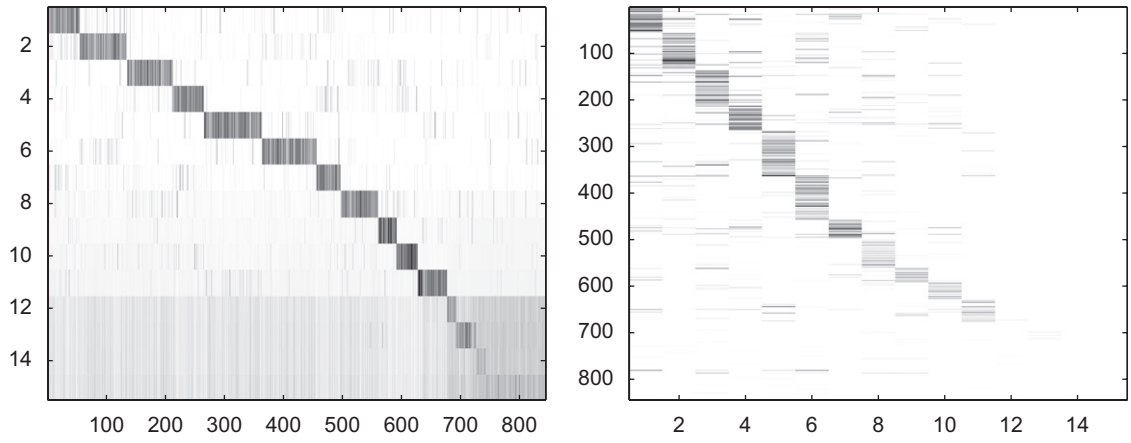


Fig. 20. The posterior expectations obtained from the bBICA decomposition. Left: $\langle \boldsymbol{a} \rangle$. Right: $\langle \boldsymbol{b} \rangle$.

point, the most probable posterior lag is obtained and these are used for reconstructing the graph of transition frequencies. These are then made binary, so that non-zero frequencies correspond to presence and zero frequencies to absence of relationships between the nodes of our graph.

Our bBICA decomposition came up with $K = 18$ components for the first-order graph and $K = 15$ for the second version of the graph. Based on these, we can reorder the nodes such that those nodes whose highest mixing coefficient is the same will be next to each other. Optionally, we can also order the components in the descending order of the sum of presence probabilities in their presence–absence pattern. Fig. 19 shows the two binary graph matrices with reordered nodes. The revealed structure is rather interesting, and it gives an entirely different alternative view from models that are based on connection frequencies (such as the one in [16]). Recall, for both graphs, the nodes and edges are untouched, only the ordering of the nodes is done using the results of bBICA. From the left-hand plot of Fig. 19 we see that bBICA separates out those nodes that form groupings and those which do not. Apparently, only about half of the participants form groupings in the first-order graph. At some closer inspection, unsurprisingly, it turns out that there are a number of 'noise' components, dominated by high probability of absence, and for the 'non-group-forming' subset of the participants, one of these noise components is the most dominant.

The right-hand figure shows results for the second version of the graph. As we can see, the amount of noise here is less. However, inevitably, noise components still do exist. In fact, one of the most important strengths of our model is to be able to separate these out so they can be identified, appropriately interpreted and the information

Table 2
Expansion of randomly picked documents from the training set

atho church rutger god sin word peopl
christian 0.52 bibl 0.34 faith 0.34 christ 0.33 jesu 0.32 accept 0.32 agre 0.31 love 0.29 speak 0.28
scriptur 0.27 truth 0.26 man 0.24 clh 0.24 teach 0.22 geneva 0.22 religion 0.22
decrypt den chip enforc escrow clipper kei encrypt
system 0.66 govern 0.60 public 0.60 secur 0.54 peopl 0.52 comput 0.40 algorithm 0.28 secret 0.27
nsa 0.26 devic 0.26 access 0.25 scheme 0.24 trust 0.23 cryptographi 0.22 pgp 0.22 privaci 0.19
man sternlight secret escrow
peopl 0.46 system 0.42 kei 0.33 encrypt 0.31 govern 0.31 public 0.31 chip 0.30 clipper 0.28 secur 0.27
comput 0.22 space 0.19 access 0.18 nasa 0.17 effect 0.15 orbit 0.15 algorithm 0.15
henri space effect
peopl 0.36 nasa 0.33 system 0.31 orbit 0.30 man 0.25 cost 0.22 launch 0.20 mission 0.18 flight 0.17
shuttl 0.16 medic 0.16 moon 0.15 solar 0.15 spacecraft 0.13 doctor 0.13 toronto 0.12
orbit space cost peopl
nasa 0.33 system 0.29 man 0.24 launch 0.20 mission 0.18 flight 0.17 shuttl 0.16 henri 0.16
moon 0.15 pat 0.15 solar 0.15 effect 0.13 spacecraft 0.13 access 0.13 spencer 0.12 toronto 0.12
space
peopl 0.40 system 0.26 nasa 0.25 orbit 0.23 man 0.20 cost 0.17 effect 0.16 launch 0.15
pat 0.15 access 0.14 mission 0.14 flight 0.13 shuttl 0.13 henri 0.12 moon 0.11 solar 0.11

The first line gives the list of words that are actually present in the document, followed by the list of 16 most likely expected additional words along with their reconstruction probabilities.

obtained from this can be appropriately used. Fig. 20 shows the actual decomposition, i.e. the matrices of posterior expectations $\langle \boldsymbol{a} \rangle$ and $\langle \boldsymbol{b} \rangle$, respectively. The former are the mixing coefficients, the latter are the beta components (characteristic presence–absence probability patterns). White corresponds to 0 and black to 1. The components are ordered w.r.t. $\sum_n \langle b_{kn} \rangle$, and the nodes are ordered by their strongest component. We see four almost entirely white components (12–15 on the right-hand plot).

In the context of social network analysis, a 'white' component means a linking pattern dominated by absence. All nodes have some non-zero mixing coefficients corresponding to white components (rows 12–15 in the left-hand plot), since it is inevitable that some links that could have been present were actually not (missed opportunities). However, we see there are also nodes whose dominant component is a white one (see columns 690–844 in rows 12–15). These are the ones for which a noise-removal will most dramatically change the mixing weights (recall, the mixing coefficients get renormalised in this operation). Removing the noise components identified from the whole network implies therefore that the links to the active components (communities) are expanded. This can be used for identifying opportunities that are not so evident otherwise, and guiding participants towards a suitable active community.

5.3. Expansion of short text messages

A final experiment considers binary coded text. That is, each text document or message is represented as a vector of size equal to the size of a common dictionary, having an entry of 1 for words that are present and an entry of 0 for words that are absent. This encoding has been used in text categorisation, in the context of Naive Bayes classification [21] and has consistently been found inferior to multinomial-based encodings. Interestingly, none of the existing literature on this subject seems to realise how noisy a binary encoding of text is. It is intuitively evident that only a small fraction of the words that could be used to express a topic are actually present in each of the documents. Moreover, some documents are really short.

We apply bBICA to analysing a subset from the 20Newsgroups collection,[5] which contains short Usenet messages from four different topics of discussion: 'sci.crypt', 'sci.med', 'sci.space' and 'soc.religion.christian'. A number of 100 documents from each newsgroup were sampled and binary term by document matrix was created using the Bow toolkit[6] over a 100 words dictionary.

Unsurprisingly, a bBICA analysis of this data consistently returns at least one blank factor. This factor is a 'semantic noise' inherent in the language. Removing the blank component has the effect of expanding the text with semantically related words. There is no objective way of quantifying this semantic relatedness; however, Tables 2 and 3 give a random sample of messages together with their expansion, as computed for examples of the training set and a small hold-out test set, respectively.

By inspection, we find the words on the expansion list are semantically strongly related to the words which are actually present in the documents. Although we cannot quantify this semantic relatedness directly, after removing the noise factor we computed the clustering error w.r.t. the true class labels and found a remarkable agreement, the

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

Table 3
Expansion of documents from a hold-out set

spirit scriptur clh church love accept agre effect peopl
god 0.51 christian 0.51 rutger 0.45 word 0.37 bibl 0.34 faith 0.34 christ 0.33 jesu 0.31
speak 0.28 truth 0.26 man 0.24 atho 0.22 teach 0.22 geneva 0.21 religion 0.21 sin 0.18
bibl clh church geneva rutger speak god christian public peopl
word 0.44 faith 0.41 christ 0.40 jesu 0.38 accept 0.37 agre 0.37 love 0.34 scriptur 0.33
truth 0.31 man 0.28 atho 0.26 teach 0.26 religion 0.26 sin 0.21 spirit 0.21 passag 0.20
pain medic patient physician doctor effect
peopl 0.53 diseas 0.41 treatment 0.38 medicin 0.37 symptom 0.31 food 0.31 med 0.30 diet 0.29
clinic 0.27 infect 0.24 syndrom 0.23 diagnos 0.22 system 0.22 accept 0.18 access 0.12 word 0.11
encrypt algorithm
peopl 0.49 system 0.36 kei 0.30 public 0.29 govern 0.29 chip 0.28 clipper 0.26 secur 0.25 comput 0.20
escrow 0.20 access 0.19 effect 0.15 pat 0.14 secret 0.13 nsa 0.13 devic 0.12 scheme 0.12 space 0.11
jesu geneva rutger christ christian
peopl 0.67 god 0.49 word 0.35 church 0.34 bibl 0.33 faith 0.32 accept 0.30 agre 0.30
love 0.27 speak 0.27 scriptur 0.26 truth 0.25 man 0.24 clh 0.23 atho 0.21 teach 0.21
syndrom symptom medicin medic diseas med doctor peopl
effect 0.68 patient 0.45 treatment 0.38 physician 0.32 food 0.31 diet 0.29 pain 0.28 clinic 0.28
infect 0.25 diagnos 0.22 system 0.21 accept 0.18 access 0.12 word 0.11 chip 0.11 agre 0.08
pgp public kei encrypt
peopl 0.50 system 0.50 govern 0.43 chip 0.42 clipper 0.40 secur 0.39 escrow 0.30 comput 0.30
access 0.22 algorithm 0.20 secret 0.20 nsa 0.19 devic 0.19 scheme 0.18 trust 0.17 cryptographi 0.16
orbit lunar spacecraft moon nasa
space 0.60 system 0.41 man 0.37 cost 0.33 launch 0.32 mission 0.28 flight 0.27 shuttl 0.26
henri 0.25 peopl 0.24 solar 0.23 spencer 0.19 toronto 0.18 vehicl 0.17 zoo 0.17 satellit 0.17

mismatch was 5.75% in average. Having detected and realised this semantic noise gives an additional insight into why binary text encodings have not been so fruitful in text categorisation in their basic form. Denoising of text data may provide interesting new avenues and could also be used e.g. for query expansion in query-based search engines.

6. Conclusions

We have devised a variational method for the factorisation of 0–1 data, employing independent beta latent densities. This model is particularly suited for denoising problems, as shown in a set of comparative experiments. We also demonstrated the use and good performance of our approach on a number novel application domains, including social network analysis and DNA fingerprint analysis. The method may have further applications. In particular, since missing value patterns are binary vectors, the method devised here could be investigated for modelling non-ignorable missing data mechanisms in conjunction with other appropriate data models being employed for the observed data.

Acknowledgements

Thanks to Xin Wang for performing the mixed-transition Markov analysis for the preprocessing of the chat sequence.

Appendix A. Variational EM solution

A.1. Inference

A.1.1. Computing $q(b_{kn})$

Taking functional derivative from

$$\mathscr{L}^{\text{bound}}(\boldsymbol{x}_n) - \sum_k \lambda_k \left(\int q(b_{kn})\, \mathrm{d}b_k - 1 \right)$$

w.r.t. the variational density function $q(b_{kn})$ and setting it to the identically null function, we obtain the optimal form of this function. The last term is a Lagrangian term, with Lagrange multipliers λ_k, to ensure proper normalisation of the obtained variational density:

$$\begin{aligned} \frac{\partial \mathscr{L}^{\text{bound}}}{\partial q(b_{kn})} &= \sum_t Q_{tn}(k|x_{tn}) \log\{b_{kn}^{x_{tn}}(1-b_{kn})^{(1-x_{tn})}\} \\ &\quad + \log B(b_{kn}|\alpha_k^0, \beta_k^0) - \log q(b_{kn}) - 1 - \lambda_k \qquad \text{(A.1)} \\ &= \log\{b_{kn}^{\sum_{t|x_{tn}=1} x_{tn} Q_{tn}(k|x_{tn}=1)} \\ &\quad \times (1-b_{kn})^{\sum_{t|x_{tn}=0} (1-x_{tn}) Q_{tn}(k|x_{tn}=0)}\} \\ &\quad + \log B(b_{kn}|\alpha_k^0, \beta_k^0) - \log q(b_{kn}) - 1 - \lambda_k \\ &= 0. \qquad \text{(A.2)} \end{aligned}$$

Isolating $\log q(b_{kn})$ and exponentiating both sides, we obtain

$$\begin{aligned} q(b_{kn}) &\propto B(b_{kn}|\alpha_k^0,\beta_k^0) b_{kn}^{\sum_{t|x_{tn}=1} x_{tn} Q_{tn}(k|x_{tn}=1)} \\ &\quad \times (1-b_{kn})^{\sum_{t|x_{tn}=0}(1-x_{tn})Q_{tn}(k|x_{tn}=0)} \\ &\propto b_{kn}^{\alpha_k^0-1}(1-b_{kn})^{\beta_k^0-1}\, b_{kn}^{\sum_{t|x_{tn}=1} x_{tn} Q_{tn}(k|x_{tn}=1)} \\ &\quad \times (1-b_{kn})^{\sum_{t|x_{tn}=0}(1-x_{tn})Q_{tn}(k|x_{tn}=0)} \\ &= b_{kn}^{\alpha_k^0+\sum_{t|x_{tn}=1} x_{tn} Q_{tn}(k|x_{tn}=1)-1} \\ &\quad \times (1-b_{kn})^{\beta_k^0+\sum_{t|x_{tn}=0}(1-x_{tn})Q_{tn}(k|x_{tn}=0)-1} \\ &\propto B(b_{kn}|\alpha_{kn},\beta_{kn}), \end{aligned} \tag{A.3}$$

where

$$\alpha_{kn} = \alpha_k^0 + \sum_{t|x_{tn}=1} x_{tn} Q_{tn}(k|x_{tn}=1), \tag{A.4}$$

$$\beta_{kn} = \beta_k^0 + \sum_{t|x_{tn}=0} (1-x_{tn}) Q_{tn}(k|x_{tn}=0). \tag{A.5}$$

Hence, the optimal free-form factorial variational posteriors are beta densities.

A.1.2. Computing Q

Solving the stationary equations from

$$\sum_n \mathscr{L}^{\text{bound}} \boldsymbol{x}_n + \sum_{n,t,x_{tn}} v_{n,t,x_{tn}} \left(\sum_k Q_{tn}(k|x_{tn}) - 1 \right)$$

yields:

$$\begin{aligned} \frac{\partial \mathscr{L}^{\text{bound}}}{\partial Q_{tn}(k|x_{tn})} &= \log a_{tk} + \langle \log b_{kn}^{x_{tn}} (1-b_{kn})^{1-x_{tn}} \rangle \\ &\quad - \log Q_{tn}(k|x_{tn}) - 1 - v_{n,t,x_{tn}} \\ &\Rightarrow Q_{tn}(k|x_{tn}) \propto a_{tk} (\mathrm{e}^{\langle \log b_{kn} \rangle})^{x_{tn}} (\mathrm{e}^{\langle \log(1-b_{kn}) \rangle})^{1-x_{tn}} \end{aligned} \tag{A.6}$$

with the normalisation being

$$\sum_{k'} a_{tk'} (\mathrm{e}^{\langle \log b_{k'n} \rangle})^{x_{tn}} (\mathrm{e}^{\langle \log(1-b_{k'n}) \rangle})^{1-x_{tn}},$$

so that indeed the constraints are satisfied.

From the above, we also have

$$Q_{tn}(k|x_{tn}=0) \propto a_{tk} (\mathrm{e}^{\langle \log(1-b_{kn}) \rangle})^{1-x_{tn}}, \tag{A.7}$$

$$Q_{tn}(k|x_{tn}=1) \propto a_{tk} (\mathrm{e}^{\langle \log b_{kn} \rangle})^{x_{tn}}, \tag{A.8}$$

which are required in (A.4) and (A.5), respectively. Using these, and making the normalisation factor explicit, $Q_{tn}(k|x_{tn})$ can also be expressed as

$$Q_{tn}(k|x_{tn}) \propto a_{tk} \left\{ \frac{x_{tn} \mathrm{e}^{\langle \log b_{kn} \rangle}}{\sum_{k'} a_{tk'} \mathrm{e}^{\langle \log b_{k'n} \rangle}} + \frac{(1-x_{tn}) \mathrm{e}^{\langle \log(1-b_{kn}) \rangle}}{\sum_{k'} a_{tk'} \mathrm{e}^{\langle \log(1-b_{k'n}) \rangle}} \right\}. \tag{A.9}$$

A.2. Estimation of the mixing parameters

To obtain a maximum likelihood estimate of the mixing matrix, we solve the stationary equation of a_{tk} from

$$\sum_n \mathscr{L}^{\text{bound}}(\boldsymbol{x}_n) + \sum_t \mu_t \left(\sum_k a_{tk} - 1 \right),$$

where μ_t are Lagrange multipliers:

$$\frac{\partial \mathscr{L}^{\text{bound}}}{\partial a_{tk}} = \sum_n Q_{tn}(k|x_{tn})/a_{tk} - \mu_t = 0. \tag{A.10}$$

Multiplying both sides by a_{tk}, we obtain

$$\sum_n Q_{tn}(k|x_{tn}) - \mu_t a_{tk} = 0 \;\; \Rightarrow \;\; a_{tk} \propto \sum_n Q_{tn}(k|x_{tn}). \tag{A.11}$$

Summing over k and using the constraint that $\sum_k a_{tk} = 1$, the normalisation factor is found to be $\sum_{k,t} Q_{tn}(k|x_{tn}) = T$.

Appendix B. Variational Bayesian solution

B.1. The evidence bound

$$\begin{aligned} &\log P(\boldsymbol{X}) \\ &= \log \iint \prod_n \left[P(\boldsymbol{x}_n|\boldsymbol{b}_n, \boldsymbol{A}) \prod_{k=1}^K p(b_{kn})\, \mathrm{d}b_{kn} \right] \\ &\quad \times \prod_{t=1}^T D(\boldsymbol{a}_t|\boldsymbol{\gamma}^0)\, \mathrm{d}\boldsymbol{a}_t \geqslant \mathscr{E}^{\text{bound}}(\boldsymbol{X}) \\ &= \sum_{n,t,k} Q_{tn}(k|x_{tn}) \{ \langle \log a_{tk} \rangle + \langle \log b_{kn}^{x_{tn}} (1-b_{kn})^{1-x_{tn}} \rangle \\ &\quad - \log Q_{tn}(k|x_{tn}) \} + \sum_{n,k} \{ \langle \log B(b_{kn}|\alpha_k^0, \beta_k^0) \\ &\quad - \log q(b_{kn}) \rangle \} + \sum_t \{ \langle \log D(\boldsymbol{a}_t|\boldsymbol{\gamma}^0) \\ &\quad - \log q(\boldsymbol{a}_t) \rangle \}, \end{aligned} \tag{B.1}$$

where D denotes the Dirichlet distribution, $\boldsymbol{A}$ stands for all mixing variables $(\boldsymbol{a}_1, \ldots, \boldsymbol{a}_t, \ldots, \boldsymbol{a}_T)$ and $\langle . \rangle$ denotes expectation w.r.t. $q(b_{kn})$ or $q(\boldsymbol{a}_t)$, as appropriate.

B.2. Computing the variational posteriors of the mixing coefficients

We take functional derivative from

$$\mathscr{E}^{\text{bound}}(\boldsymbol{X}) - \sum_t \tilde{\lambda}_t \left(\int q(\boldsymbol{a}_t)\, \mathrm{d}\boldsymbol{a}_t - 1 \right)$$

w.r.t. the variational distribution $q_t(\boldsymbol{a})$, where $\tilde{\lambda}_t$ are Lagrange multipliers to ensure proper normalisation. This is

$$\begin{aligned}\frac{\partial}{\partial q(\boldsymbol{a}_t)} &= \sum_n \sum_k Q_{tn}(k|x_{tn}) \log a_{tk} + \log D(\boldsymbol{a}_t|\boldsymbol{\gamma}^0) \\ &\quad - \log q(\boldsymbol{a}_t) - 1 - \tilde{\lambda}_t \\ &= \sum_k \log a_{tk}^{\sum_n Q_{tn}(k|x_{tn})} + \log D(\boldsymbol{a}_t|\boldsymbol{\gamma}^0) \\ &\quad - \log q(\boldsymbol{a}_t) - 1 - \tilde{\lambda}_t. \end{aligned} \tag{B.2}$$

Isolating $\log q(\boldsymbol{a}_t)$ and exponentiating both sides, we get

$$q(\boldsymbol{a}_t) \propto D(\boldsymbol{a}_t|\boldsymbol{\gamma}^0) \prod_k a_{tk}^{\sum_n Q_{tn}(k|x_{tn})} \tag{B.3}$$

$$\begin{aligned} &\propto \prod_k a_{tk}^{\gamma_k^0 - 1} \prod_k a_{tk}^{\sum_n Q_{tn}(k|x_{tn})} \\ &= \prod_k a_{tk}^{\gamma_k^0 + \sum_n Q_{tn}(k|x_{tn}) - 1} \\ &\propto D\left(\boldsymbol{a}_t|\gamma_k^0 + \sum_n Q_{tn}(k|x_{tn})\right). \end{aligned} \tag{B.4}$$

Hence, the optimal variational posterior mixing distributions are Dirichlet densities, with variational parameters

$$\gamma_{tk} \equiv \gamma_k^0 + \sum_n Q_{tn}(k|x_{tn}). \tag{B.5}$$

B.3. Computing Q

The computation of $Q_{tn}(k|x_{tn})$ follows the same route as before, and formally the only difference is that now instead of the parameters a_{tk} we have $e^{\langle \log a_{tk} \rangle}$ throughout.

Appendix C. Posterior data reconstruction

The posterior probability that a data entry is reconstructed as a 1 can be expressed using the Bernoulli likelihood and the model posteriors. For the model estimated by the variational Bayes procedure, this is the following:

$$\begin{aligned} P(\hat{x}_{tn} = 1|\boldsymbol{X}, \boldsymbol{u}) &= \iint P(\hat{x}_{tn} = 1|\boldsymbol{b}, \boldsymbol{a}_t) \\ &\quad \times \prod_k q(b_{kn})\, \mathrm{d}b_k q(\boldsymbol{a}_t|\boldsymbol{u})\, \mathrm{d}\boldsymbol{a}_t \qquad \text{(C.1)} \\ &= \iint \sum_k a_{tk} b_k q(b_{kn})\, \mathrm{d}b_k\, q(\boldsymbol{a}_t|\boldsymbol{u})\, \mathrm{d}\boldsymbol{a}_t \qquad \text{(C.2)} \\ &= \sum_k \langle a_{tk}|\boldsymbol{u}\rangle \langle b_k\rangle. \qquad \text{(C.3)} \end{aligned}$$

References

[1] H. Attias, A variational Bayesian framework for graphical models, in: S. Solla, T. Leen, K.R. Müller (Eds.), Advances in Neural Information Processing Systems, vol. 12, MIT Press, Cambridge, MA, USA, 2000, pp. 209–215.

[2] M.J. Beal, Z. Ghahramani, Variational Bayesian learning of directed graphical models with hidden variables, Bayesian Stat. 1 (4) (2006).

[3] A. Berchtold, A. Raftery, The mixture transition distribution model for high-order Markov chains and non-Gaussian time-series, Stat. Sci. 17 (3) (2002) 328–356.

[4] J.M. Bernardo, A.F.M. Smith, Bayesian Theory, Wiley, New York, 2001.

[5] D.M. Blei, A.Y. Ng, M.I. Jordan, Latent Dirichlet allocation, J. Mach. Learn. Res. 3 (5) (2003) 993–1022.

[6] W. Buntine, A. Jakulin, Applying discrete PCA in data analysis, in: Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence, 2004, pp. 59–66.

[7] C. Constantinopoulos, M.K. Titsias, A. Likas, Bayesian feature and model selection for Gaussian mixture models, IEEE Trans. Pattern Anal. Mach. Intell. 28 (6) (2006) 1013–1018.

[8] K.I. Diamantaras, T. Papadimitriou, Blind deconvolution of multi-input single-output systems with binary sources, IEEE Trans. Signal Process. 54 (10) (2006) 3720–3731.

[9] T. Fawcett, ROC graphs: notes and practical considerations for researchers, Technical Report, HP Laboratories, MS 1143, Palo Alto CA, USA, April 2004.

[10] M.S. Handcock, A.E. Raftery, Model-based clustering for social networks, J. R. Stat. Soc. A 170 (Part 2) (2007) 1–22.

[11] J. Himberg, A. Hyvärinen, Independent component analysis for binary data: an experimental study, in: Proceedings of the ICA2001, 2001, pp. 552–556.

[12] A. Hyvärinen, J. Karhunen, E. Oja, Independent Component Analysis, Wiley, New York, 2001.

[13] M. Jordan, Z. Ghahramani, T. Jaakkola, L. Saul, An introduction to variational methods for graphical models, in: M. Jordan (Ed.), Learning in Graphical Models, MIT Press, Cambridge, MA, USA, 1999, pp. 105–161.

[14] T.P. Jung, S. Makeig, C. Humphries, T.W. Lee, M.J. McKeown, V. Iragui, T.J. Sejnowski, Removing electroencephalographic artifacts by blind source separation, Psychophysiology 37 (2000) 163–178.

[15] A. Kabán, E. Bingham, ICA-based binary feature construction, in: Proceedings of the 6th International Conference on Independent Component Analysis and Blind Source Separation (ICA06), Lecture Notes in Computer Science, vol. 3889, Springer, Berlin, 2006, pp. 140–148.

[16] A. Kabán, X. Wang, Deconvolutive clustering of Markov states, in: Proceedings of the ECML'06, 2006, pp. 246–257.

[17] H. Lappalainen, Ensemble learning for independent component analysis, in: Proceedings of the International Workshop on Independent Component Analysis and Signal Separation (ICA'99), Aussois, France, 1999, pp. 7–12.

[18] D.D. Lee, H.S. Seung, Algorithms for non-negative matrix factorization, Adv. Neural Inf. Process. Syst. (2000) 556–562.

[19] Y. Li, A. Cichocki, L. Zhang, Blind separation and extraction of binary sources, IEICE Trans. Fund. Electron. Commun. Comput. Sci. E86-A (3) (2003) 580–589.

[20] B. Marlin, Modeling user rating profiles for collaborative filtering, in: Proceedings of the Neural Information Processing Systems, 2003.

[21] A. McCallum, K. Nigam, A comparison of event models for Naive Bayes text classification, in: Proceedings of AAAI/ICML-98 Workshop on Learning for Text Categorization, 1998, pp. 41–48.

[22] P. McCullagh, J.A. Nelder, Generalized Linear Models, Chapmann & Hall, London, 1983.

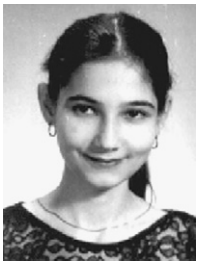
[23] M. Rees, from 'research quotes' ⟨http://www.c2i.ntu.edu.sg/AI+CI/Humor/researchquotes.html⟩.

[24] E. Saund, A multiple cause model for unsupervised learning, Neural Comput. 7 (1995) 51–71.

[25] T. Singliar, M. Hauskrecht, Noisy-OR component analysis and its application to link analysis, J. Mach. Learn. Res. 7 (2006) 2189–2213.

[26] A.I. Schein, L.K. Saul, L.H. Ungar, A generalised linear model for principal component analysis of binary data, in: Proceedings of the 9th International Workshop on Artificial Intelligence and Statistics, 2003.

[27] M.E. Tipping, Probabilistic visualisation of high dimensional data, Adv. Neural Inf. Process. Syst. (1999) 592–598.
[28] J.D. Wilbur, J.K. Ghosh, C.H. Nakatsu, S.M. Brouder, R.W. Doerge, Variable selection in high-dimensional multivariate binary data with application to the analysis of microbial community DNA fingerprints, Biometrics 58 (2002) 378–386.



Ata Kaban received the B.Sc. degree with honours (1999) in computer science from the University 'Babes-Bolyai' of Cluj-Napoca, Romania, and the Ph.D. degree in computer science (2001) from the University of Paisley, UK. She is a lecturer in the School of Computer Science of the University of Birmingham. Her current interests concern probabilistic modelling, machine learning and their applications to automated data analysis.

Prior to her career in computer science, she received the B.A. degree in musical composition (1994) and the M.A. (1995) and Ph.D. (1999) degrees in musicology from the Music Academy 'Gh. Dima' of Cluj-Napoca, Romania.

Ella Bingham was born in Loimaa, Finland, in 1973. She received her M.Sc. degree in engineering physics and mathematics at Helsinki University of Technology in 1998, and her D.Sc. (Tech) degree in computer science at Helsinki University of Technology in 2003. She is currently at Helsinki Institute for Information Technology, Basic Research Unit, located at the University of Helsinki. Her research interests include data mining and machine learning.