Arto Klami and Samuel Kaski. 2008. Probabilistic approach to detecting dependencies between data sets. Neurocomputing, to appear.

Probabilistic approach to detecting dependencies between data sets

Arto Klami and Samuel Kaski

Helsinki Institute for Information Technology and

Adaptive Informatics Research Centre

Helsinki University of Technology

P.O.Box 5400, FI-02015 TKK, Finland

Abstract

We study data fusion under the assumption that data source-specific variation is irrelevant and only shared variation is relevant. Traditionally the shared variation has been sought by maximizing a dependency measure, such as correlation of linear projections in Canonical Correlation Analysis. In this traditional framework it is hard to tackle overfitting and model order selection, and thus we turn to probabilistic generative modeling which makes all tools of Bayesian inference applicable. We introduce a family of probabilistic models for the same task, and present conditions under which they seek dependency. We show that probabilistic CCA is a special case of the model family, and derive a new dependency-seeking clustering algorithm as another example. The solution is computed with variational Bayes.

Key words: Canonical correlation analysis; clustering; data fusion; mutual dependency; probabilistic modeling; variational Bayes

1 Introduction

We study the task of modeling dependencies between two data sets of co-occurring or paired samples $(\mathbf{x}, \mathbf{y})$. In other words, the task is to find what is shared by, or statistically in common between $\mathbf{x}$ and $\mathbf{y}$. The underlying assumption is that variation within either data set alone is more noisy, or at least less interesting than the variation that is in common. Example tasks include translation where the $\mathbf{x}$ and $\mathbf{y}$ are sentences in different languages, or analysis of measurement data from two different kinds of noisy sensors, such as different gene expression array platforms, that measure the same system.

This task has been classically solved by Canonical Correlation Analysis (CCA) [8,9], or more recently by other methods that maximize mutual information [4,5,16,18]. Mutual information measures deviation from independence and is hence arguably a very good objective function for finding dependencies. Unfortunately it is defined for distributions and not data sets, and hence cannot handle well the uncertainty stemming from small size of data sets. Estimation of mutual information is particularly difficult because of the "large p, small n" problem which is commonplace in bioinformatics, for instance, where the dimensionality p may be large although the number of samples n is small. Alternative Bayes factor-based dependency measures have been proposed for this task [11], but even they are so far unable to take all uncertainties into account.

The underlying principle behind the traditional methods is to transform the original data into more compact representations, for which the dependency is maximized. Usually the remaining data set-specific variation in the data

is ignored and not modeled at all, which causes difficulties in assessing the quality of the solution and in avoiding overfitting. We will approach the same problem from the opposite direction, trying to build for the data collection a generative description that contains both shared and data set-specific effects. This leads us to Bayesian generative modeling of joint distributions, in this case of $p(\mathbf{x}, \mathbf{y})$, which is a traditional well-justified framework for modeling (small) data sets. There is no reason in general, however, to expect a model of joint distributions to focus on dependencies. In this paper we introduce ways of building joint density models capable of detecting dependencies between data sources.

We introduce a generic model structure for data fusion applications, and show how applying standard Bayesian machinery (see e.g. [7]) to models having that structure makes them find dependencies between data sets. Special cases work analogously to the classical methods, but are still generative models with all their advantages. In particular, we show how the proposed model structure leads to a recent probabilistic interpretation of CCA [3] when certain restrictive assumptions, namely linearity of projections and Gaussianity of distributions, are made. We then proceed to introduce a practical clustering method for the same task, and treat it in a full-Bayesian way using the variational approximation (see e.g. [10]). The purpose is to demonstrate how the generative modeling approach leads to practical improvements.

The connection between classical CCA and a probabilistic variant [3] is that the maximum likelihood solution of the latter is equivalent to the former. The connection between dependency measures and likelihood has been pointed out in special cases also before, for example by [5] in the context of discrete variables. They were able to turn maximization of mutual information in a

co-clustering task into a maximum likelihood problem by assuming that the marginal densities of the clusters are known exactly. We extend the idea to non-discrete data, where it is not feasible to assume known marginal densities, by assuming flexible models for the data set-specific variation. We show that introducing such sources of variation into the model and integrating them out allows us to use likelihood more generally as a score function for fitting dependency-seeking models.

2 Generative model for finding dependencies between data sets

We define the task of finding dependencies as detecting a shared signal between two measurements [1] . The measurements are assumed to be generated as

$$\begin{aligned} \mathbf{x} &= \mathbf{f}(\mathbf{z}|\mathbf{W}_x) + \mathbf{g}(\mathbf{z}_x|\mathbf{B}_x) + \boldsymbol{\epsilon}_x, \\ \mathbf{y} &= \mathbf{f}(\mathbf{z}|\mathbf{W}_y) + \mathbf{g}(\mathbf{z}_y|\mathbf{B}_y) + \boldsymbol{\epsilon}_y, \end{aligned} \tag{1}$$

where the $\boldsymbol{\epsilon}_x$ and $\boldsymbol{\epsilon}_y$ denote noise and the $\mathbf{f}(\cdot)$ and $\mathbf{g}(\cdot)$ are deterministic functions that transform the latent signals $\mathbf{z}$ to the observation space. $\mathbf{W} = [\mathbf{W}_x; \mathbf{W}_y]$, $\mathbf{B} = [\mathbf{B}_x; \mathbf{B}_y]$ are parameters of these functions. In other words, it is assumed that the observed data $\mathbf{x}$ depends on two latent signals. The signal $\mathbf{z}$ is shared with data $\mathbf{y}$ and the signal $\mathbf{z}_x$ is not. The two signals add to form the actual observations.

The data fusion problem is to find the shared latent variables $\mathbf{z}$ given $\mathbf{x}$ and/or $\mathbf{y}$, as well as the parameters $\mathbf{W}_x$ and $\mathbf{W}_y$ of the mappings from $\mathbf{z}$ to the observed data. The latent variables are the fused output, whereas the parameters

[1] The formulation extends directly to more than two measurements, but the formulas are presented for the simplest case throughout the paper

are used to interpret the result. In general, this requires estimating $\mathbf{B}_x$ and $\mathbf{B}_y$, but $\mathbf{z}_x$ and $\mathbf{z}_y$ need not be explicitly constructed.

For solving the data fusion problem we will use probabilistic modeling. Bayesian generative modeling gives good tools for controlling the model complexity and for avoiding overfitting to small data sets, which are extremely difficult tasks in the traditional approaches.

We will starting by giving a full-Bayesian treatment only to the latent variables $\mathbf{z}$, $\mathbf{z}_x$, and $\mathbf{z}_y$. The rest of the parameters will be optimized by maximizing the likelihood of the parameters, given the set of observations and the model (1). We need to specify prior distributions for all of the latent variables, $\mathbf{z}$, $\mathbf{z}_x$ and $\mathbf{z}_y$, as well as the noise distributions. Additionally, as in all modeling, we will need to specify model families, here for $\mathbf{f}(\cdot)$ and $\mathbf{g}(\cdot)$. The resulting model is illustrated as a graphical model in Figure 1.

The fundamental task in the data fusion problem is to solve for the assumed shared signal $\mathbf{z}$. The naive approach would be to find all the parameters and latent variables in (1) by maximizing the likelihood, and only consider the shared part when interpreting the results. However, the computation would be inefficient and, as discussed later, would not necessarily find the correct signal. Instead, we will use the rigorous Bayesian way of treating the remaining latent variables as nuisance parameters and marginalize them out. After marginalizing out the $\mathbf{z}_x$ and $\mathbf{z}_y$ we still need to optimize the parameters $\mathbf{W}$ and $\mathbf{B}$, and as a final result we are interested in the posterior distribution of $\mathbf{z}$ given $\mathbf{x}$ and/or $\mathbf{y}$. That quantity provides the best possible estimate of the latent signal, based on the available observations.

Marginalization over $\mathbf{z}_x$ (and similarly for $\mathbf{z}_y$) means performing the inte-

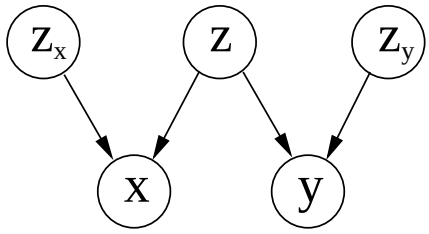

Fig. 1. Graphical representation of the generative model structure used to detect dependencies. The model assumes a shared signal $\mathbf{z}$ between the two observed variables $\mathbf{x}$ and $\mathbf{y}$, as well as signals $\mathbf{z}_x$ and $\mathbf{z}_y$ that are specific to each data set.

gration $\int p(\mathbf{z}, \mathbf{z}_x, \mathbf{z}_y, \mathbf{x}, \mathbf{y}) d\mathbf{z}_x$, which may in general be difficult but here simplifies to $p(\mathbf{z})p(\mathbf{z}_y)p(\mathbf{y}|\mathbf{z}, \mathbf{z}_y) \int p(\mathbf{x}|\mathbf{z}, \mathbf{z}_x)p(\mathbf{z}_x)d\mathbf{z}_x$ because of the independence assumptions given in Figure 1. Furthermore, $\mathbf{z}$ is constant in the integration, and we can use $p(\tilde{\mathbf{x}}|\mathbf{z}_x)$ in place of $p(\mathbf{x}|\mathbf{z}, \mathbf{z}_x)$ by denoting $\tilde{\mathbf{x}} = \mathbf{x} - \mathbf{f}(\mathbf{z}|\mathbf{W}_x)$. The additive nature of the model thus makes the marginalization tractable for a wide spectrum of model families, as the marginalization only involves the terms $\mathbf{g}(\mathbf{z}_x|\mathbf{B}_x)$ and $\boldsymbol{\epsilon}_x$.

3 Canonical correlation analysis

Canonical correlation analysis (CCA) [8,9] is a classical linear model for finding dependencies between two data sets. It is formulated as a search for the linear transformations $\mathbf{U}_x$ and $\mathbf{U}_y$ such that each dimension of $\mathbf{U}_x\mathbf{x}$ correlates maximally with the corresponding dimension of $\mathbf{U}_y\mathbf{y}$. CCA thus finds what the two data sets have in common by explicitly maximizing the correlation. The solution can be effectively computed by solving a certain generalized eigenvalue problem. The solution is a unique global optimum but the method is known to overfit to small data sets (see e.g [8] for an overview of classical CCA, including a kernel-based extension). In this section we show how

the probabilistic interpretation of CCA [3] can be derived from the model structure (1) (Figure 1). This derivation makes possible extensions; a sample extension is given in the next Section. Throughout the derivation we consider only the formulas for the x-space; the y-space is completely analogous.

As CCA works with linear projections the $\mathbf{f}$ and $\mathbf{g}$ in (1) need to be linear as well, giving $\mathbf{f}(\mathbf{z}|\mathbf{W}_x) = \mathbf{W}_x\mathbf{z}$ and $\mathbf{g}(\mathbf{z}_x|\mathbf{B}_x) = \mathbf{B}_x\mathbf{z}_x$. Furthermore, we make the common choice of the noise being independent Gaussian with equal variance in each dimension, $\boldsymbol{\epsilon}_x = N(0, \sigma_x^2\mathbf{I})$. The prior distributions of the $\mathbf{z}$, $\mathbf{z}_x$, and $\mathbf{z}_y$ are all assumed to be Gaussian with zero mean and unit covariance matrix, analogously to the probabilistic interpretation of principal component analysis (PCA) [15,17].

In summary, in the generative model with the above assumptions we have

$$\begin{aligned} \mathbf{x} &\sim N(\mathbf{W}_x\mathbf{z} + \mathbf{B}_x\mathbf{z}_x, \sigma_x^2\mathbf{I}), \\ \mathbf{z}, \mathbf{z}_x &\sim N(0, \mathbf{I}), \end{aligned}$$

and in order to infer $\mathbf{z}$ we need to marginalize out the latent variables $\mathbf{z}_x$. As explained in the previous section the marginalization only involves an integral over $p(\tilde{\mathbf{x}}|\mathbf{z}_x)p(\mathbf{z}_x)$, where $\tilde{\mathbf{x}} = \mathbf{x} - \mathbf{W}_x\mathbf{z}$.

As the prior $p(\mathbf{z}_x)$ is Gaussian and it is multiplied with a linear term we can integrate $\mathbf{z}_x$ out analytically, obtaining $\tilde{\mathbf{x}} \sim N(0, \mathbf{B}_x\mathbf{B}_x^T + \sigma_x^2\mathbf{I})$ (see e.g. [15] for general cases of integrals in Gaussian latent variable models). This is a normal distribution where the covariance matrix has been parameterized to have rank $d_{z_x} + 1$. Here d_{z_x} is the number of dimensions in $\mathbf{z}_x$ (assuming rows of $\mathbf{B}_x$ are linearly independent).

If we assume that $d_{z_x} + 1$ matches the dimensionality of $\mathbf{x}$ we can change the

parameterization of the generative model to $\tilde{\mathbf{x}} \sim N(0, \mathbf{\Psi}_x)$. Here $\mathbf{\Psi}_x$ is a freely parameterized positive definite matrix, and the parameterizations are equivalent since both provide a non-constrained covariance matrix. This special case is equivalent to a recent probabilistic interpretation of Canonical Correlation Analysis (CCA) [3], as will be shown next.

Doing the same marginalization for $\mathbf{z}_y$, again assuming that the dimensionality of $\mathbf{z}_y$ is large enough to enable a non-constrained covariance matrix $\mathbf{\Psi}_y$, leads to the generative model

$$\begin{aligned} \mathbf{z} &\sim N(0, \mathbf{I}), \\ \mathbf{x}|\mathbf{z} &\sim N(\mathbf{W}_x\mathbf{z}, \mathbf{\Psi}_x), \\ \mathbf{y}|\mathbf{z} &\sim N(\mathbf{W}_y\mathbf{z}, \mathbf{\Psi}_y). \end{aligned}$$

This is exactly the model proposed in [3] for interpreting CCA probabilistically. As shown in [3], the maximum likelihood estimates of $\mathbf{W}_x$ and $\mathbf{W}_y$ are related to the classical CCA solution by $\mathbf{W}_x = \mathbf{\Sigma}_x\mathbf{U}_x\mathbf{Q}_x$ and $\mathbf{W}_y = \mathbf{\Sigma}_y\mathbf{U}_y\mathbf{Q}_y$, where $\mathbf{\Sigma}_x$ and $\mathbf{\Sigma}_y$ are the empirical covariance matrices. $\mathbf{Q}_x$ and $\mathbf{Q}_y$ are arbitrary matrices with spectral normal smaller than one, such that $\mathbf{Q}_x\mathbf{Q}_y^T = \mathbf{P}$, where $\mathbf{P}$ is a diagonal matrix that contains the canonical correlations.

More importantly, the expectations of the latent variables $\mathbf{z}$ given the data lie in the subspace found by CCA. As explained in Section 2, the posterior distribution of the shared latent signal is the quantity we are interested in, and the expectation is the best point estimate of that distribution. The exact connection to CCA holds for $\mathrm{E}[\mathbf{z}|\mathbf{x}]$ and $\mathrm{E}[\mathbf{z}|\mathbf{y}]$, and while these are interesting quantities as such the probabilistic model gives also a better estimate of the shared signal as $\mathrm{E}[\mathbf{z}|\mathbf{x}, \mathbf{y}]$. It is more accurate since it utilizes all the available information. The equivalent combination of information in the case of

traditional CCA would be the mean of the canonical scores, a quantity that is not commonly used in CCA analysis. It has been pointed out recently [14], however, that when using CCA as a preprocessing method the mean of the canonical scores is a feature that extracts common variation from a collection of data sets.

The original probabilistic formulation has a rotational ambiguity in the sense that the actual CCA components are not revealed. In [1] a straightforward way is proposed for solving the true components by a post-processing step, which makes it possible to use the probabilistic method in a way completely analogous to classical CCA.

Traditionally CCA has been solved directly with linear algebra, and the probabilistic version can be solved using an expectation maximization (EM) algorithm presented in [3]. As discussed above, that solution implicitly assumes that the dimensionalities of the $\mathbf{z}_x$ and $\mathbf{z}_y$ are sufficiently high to produce a non-constrained covariance matrix. The model structure in Figure 1 does not require making this assumption, however, and in Table 1 we give a more general EM algorithm for linear projections. The algorithm includes a step that follows directly from the theory, that is a step that updates the $\mathbf{W}_x$, $\mathbf{W}_y$, and $\mathbf{z}$ after having marginalized the $\mathbf{z}_x$ and $\mathbf{z}_y$ out, but also an additional step that marginalizes the $\mathbf{z}$ out to enable estimation of the $\mathbf{B}_x$ and $\mathbf{B}_y$.

The computational complexity of a single iteration of the algorithm is cubic in the dimensionalities of the data sets, and linear in the number of samples. This equals the computational complexity of traditional algorithms used for solving CCA, but in practice the iterative EM algorithm is considerably slower. It is still easily computable for relatively large data sets, and more general since it

Table 1

EM algorithm for optimizing the extended probabilistic CCA repeats the two steps until convergence. The second step can be repeated a few times in a row to improve the convergence of the data set-specific models, avoiding unnecessary use of the parameters $\mathbf{W}_x$ and $\mathbf{W}_y$ to model effects specific to each data.

1. • Assume that $\mathbf{B}_x$ and $\mathbf{B}_y$ are fixed, and marginalize over $\mathbf{z}_x$ and $\mathbf{z}_y$ to get $\mathbf{\Psi}_x = \mathbf{B}_x\mathbf{B}_x^T + \sigma_x^2\mathbf{I}$ and $\mathbf{\Psi}_y = \mathbf{B}_y\mathbf{B}_y^T + \sigma_y^2\boldsymbol{I}$.

• Update the parameters $\mathbf{W} = [\mathbf{W}_x; \mathbf{W}_y]$ using

$$\mathbf{W} = \mathbf{\Sigma}\mathbf{A}^T\left(\mathbf{M} + \mathbf{A}\mathbf{\Sigma}\mathbf{A}^T\right)^{-1}.$$

Here $\mathbf{M} = \left(\mathbf{I} + \mathbf{W}^T\mathbf{\Psi}^{-1}\mathbf{W}\right)^{-1}$, $\mathbf{A} = \mathbf{M}\mathbf{W}^T\mathbf{\Psi}^{-1}$, and $\mathbf{\Psi}$ is a block-diagonal matrix that consists of $\mathbf{\Psi}_x$ and $\mathbf{\Psi}_y$. The $\mathbf{\Sigma}$ is the joint sample covariance matrix.

2. • Marginalize over $\mathbf{z}$ to get $\mathbf{\Psi}_x = \mathbf{W}_x\mathbf{W}_x^T + \sigma_x^2\mathbf{I}$.

• Update $\mathbf{B}_x$ with

$$\mathbf{B}_x = \mathbf{\Sigma}_x\mathbf{A}_x^T\left(\mathbf{M}_x + \mathbf{A}_x\mathbf{\Sigma}_x\mathbf{A}_x^T\right)^{-1},$$

where $\mathbf{M}_x = \left(\mathbf{I} + \mathbf{B}_x^T\mathbf{\Psi}_x^{-1}\mathbf{B}_x\right)^{-1}$, $\mathbf{A}_x = \mathbf{M}_x\mathbf{B}_x^T\mathbf{\Psi}_x^{-1}$, and $\mathbf{\Sigma}_x$ is the sample covariance of $\mathbf{x}$.

• Update σ_x^2 using

$$\sigma_x^2 = \frac{1}{d_x}\text{trace}\left(\mathbf{\Sigma}_x - \mathbf{\Sigma}_x\mathbf{A}_x^T\mathbf{B}_x^T - \mathbf{W}_x\mathbf{W}_x^T\right),$$

where d_x is the dimensionality of $\mathbf{x}$, and $\mathbf{B}_x$ is the new value just updated.

• Repeat the above two substeps for parameters related to $\mathbf{y}$, replacing all subscripts x with y.

provides also the projections $\mathbf{B}_x$ and $\mathbf{B}_y$.

3.1 The role of the data set-specific latent signals

In the previous section the connection to CCA was derived for the special case of the latent signal space for the data set-specific variation having full dimensionality. It is important to realize that this is the only case where the connection to CCA holds. More generally, the proposed model structure (1) focuses entirely on detecting the shared latent signal $\mathbf{z}$ only if the latent signals $\mathbf{z}_x$ and $\mathbf{z}_y$ have full-dimensionality. In other words, the part of the model (1) that is specific to $\mathbf{x}$, $\mathbf{g}(\mathbf{z}_x|\mathbf{B}_x)$, needs to be capable of modeling any non-shared variation within $\mathbf{x}$.

If that requirement is not satisfied the model may still be a good generative description of the data collection, but it will not detect the shared effect $\mathbf{z}$ correctly. As an extreme example we may consider a special case where the dimensionality of $\mathbf{z}_x$ and $\mathbf{z}_y$ is zero, meaning that the data set-specific part is omitted completely. The model then reduces to a trivial latent variable model for the concatenation of $\mathbf{x}$ and $\mathbf{y}$, and we get essentially a probabilistic PCA [17]. The only difference is that both data spaces now have a separate noise parameter. In practice the model will fail to discover dependencies between data sets, since it uses the assumed shared latent signal to describe also variation that is specific to either data alone, the reason being that the single latent signal needs to describe the whole data generation process and has no other way for modeling the data set-specific variation.

The requirement also clarifies the need for marginalizing over the $\mathbf{z}_x$ and $\mathbf{z}_y$ instead of simply finding the maximum likelihood estimate also for them. As the models are required to be flexible there would be serious risk for overfitting

these less interesting signals, leading to the opposite problem: The model might prefer describing also some of the shared effects by the latent signals $\mathbf{z}_x$ and $\mathbf{z}_y$, not correctly detecting all the dependencies in $\mathbf{z}$.

4 Dependency-seeking clustering

In Section 3 we showed how a classical dependency maximization method CCA can be derived from (1). In the case of CCA the advantage of the probabilistic approach is in enabling robust analysis of small data sets, whereas for large sets it may in practice still be easier to use classical CCA instead. Changing and relaxing the modeling assumptions gives new models which still detect shared signals. Next we give an example demonstrating that (1) is not merely a re-interpretation of a classical method, but opens up possibilities for novel methods. In particular, these methods may not have direct traditional counterparts.

An interesting variant is obtained by assuming that the shared signal forms clusters. This is particularly useful for two reasons: with clusterings it is possible to approximate nonlinear functions, and clusterings are readily interpretable in practical data analyses. We will derive a model which assumes a clustered shared signal and Gaussian data set-specific sources. Alternatively, the data set-specific signal could be assumed to be clustered, although such a model would be computationally more difficult.

The clustering effect can be achieved by changing the prior distribution of $\mathbf{z}$. Instead of drawing $\mathbf{z}$ from a Gaussian we use a multinomial distribution (making it a scalar z instead of a vector), assuming that there is a discrete set of

possible values for the shared signal. Otherwise we use the same distributions as in Section 3, namely $\mathbf{z}_x, \mathbf{z}_y \sim N(0, \mathbf{I})$ and $\boldsymbol{\epsilon}_x, \boldsymbol{\epsilon}_y \sim N(0, \sigma^2\mathbf{I})$. The function $\mathbf{f}(z|\mathbf{W}_x)$ will here be $\mathbf{W}_x\tilde{\mathbf{z}}$, where $\tilde{\mathbf{z}}$ is the latent signal decoded as a binary vector where the zth value is one and all others are zero (i.e., the value of z picks a single column from the matrix $\mathbf{W}_x$), and $\mathbf{g}(\mathbf{z}_x|\mathbf{B}_x) = \mathbf{B}_x\mathbf{z}_x$ like in CCA.

Following the guidelines in Sections 2 and 3.1 we again assume that the data set-specific signals $\mathbf{z}_x$ and $\mathbf{z}_y$ are of a sufficiently high dimension and marginalize over them. As this part of the model is identical to the CCA model in Section 3, we can readily utilize the results from there. The result of the marginalization is $\mathbf{x} \sim N(\boldsymbol{\mu}_x^k, \mathbf{B}_x\mathbf{B}_x^T + \sigma_x^2\mathbf{I})$, where $\boldsymbol{\mu}_x^k = \mathbf{W}_x\tilde{\mathbf{z}}$, the mean parameter chosen by the latent variable z having value k. The assumption of full dimensionality for the data set-specific signals again gives equivalent parameterization in the form $\mathbf{x} \sim N(\boldsymbol{\mu}_x^k, \boldsymbol{\Psi}_x)$, and we can directly write the final clustering model as

$$\begin{aligned} z &\sim \text{Mult}(\boldsymbol{\alpha}), \\ [\mathbf{x}; \mathbf{y}]|z &\sim N(\boldsymbol{\mu}^k, \boldsymbol{\Psi}). \end{aligned} \tag{2}$$

Here $\boldsymbol{\Psi} = [\boldsymbol{\Psi}_x, \mathbf{0}; \mathbf{0}, \boldsymbol{\Psi}_y]$, a block-diagonal matrix consisting of the $\boldsymbol{\Psi}_x$ and $\boldsymbol{\Psi}_y$, and $\boldsymbol{\mu}^k = [\boldsymbol{\mu}_x^k; \boldsymbol{\mu}_y^k]$. The multinomial for z is parameterized by α to allow clusters to have different weights.

In summary, the model is a normal mixture model for data where the two feature vectors have been concatenated, with the restriction that the covariance of the clusters has a block-diagonal structure. The intuitive approach to clustering such data would be to use a full covariance matrix. It would in this case lead to individual clusters modeling also some of the dependencies be-

tween data sets, and even though that solution might be better in terms of the likelihood it would still be worse for making inferences on the dependencies. In the other extreme where the covariance matrix would be restricted to be completely diagonal the model would use the cluster structure to model also within-data variation, again losing some of the dependencies. This latter effect is analogous to CCA reducing to PCA if the data set-specific components are removed, as mentioned in Section 3.1.

Note that the formulation suggests that the covariance matrix should be restricted also in cases where the variables in both data sets are expected to be correlated (for example, due to being measurements of the same objects with different sensors), as long as the task is to detect dependencies. This is because we specifically want to capture the real link between the two data sets into the cluster structure, instead of in the within-cluster covariance. This contradicts the traditional view of Bayesian data analysis that all prior information should be included in the model structure as well as possible.

It is additionally worth noting that the model structure as written in (1) suggests that all clusters should have the same covariance matrix. This can be relaxed by allowing $\mathbf{g}(\cdot)$ to depend also on z, which arguably makes more sense in the case of a clustering model. The only change in the model would be that $\mathbf{\Psi}$ would then be replaced by $\mathbf{\Psi}^k$, an individual parameter for each cluster. We could equivalently have written (1) in that form, but left the dependency out to clarify the CCA derivation, as the independency assumption would have been made anyway for the CCA model. In all of the clustering experiments in this paper the covariance matrices have been defined separately for each cluster.

4.1 Variational Bayes for the clustering model

It would be straightforward to derive an EM algorithm to optimize (2). Here we, however, treat the estimation process in a fully Bayesian way to get the full advantages of the generative approach. Rigorous Bayesian treatment allows choosing the model complexity (i.e., the number of clusters), and the result incorporates knowledge of the uncertainty of the obtained parameters. In particular, the issue of overfitting that was one of the original reasons to pursue the generative approach can be solved much more effectively than by using maximum likelihood as the model fitting criterion.

We start by defining the model as a fully Bayesian version with priors for the model parameters, as follows:

$$\begin{aligned}
\boldsymbol{\alpha}|\lambda^0 &\sim Dir(\lambda^0) \\
z|\boldsymbol{\alpha} &\sim Mult(\boldsymbol{\alpha}) \\
\boldsymbol{\Gamma}_x^k|\nu_x^0, \boldsymbol{\Psi}_x^0 &\sim W(\nu_x^0, \boldsymbol{\Psi}_x^0) \\
\boldsymbol{\Gamma}_y^k|\nu_y^0, \boldsymbol{\Psi}_y^0 &\sim W(\nu_y^0, \boldsymbol{\Psi}_y^0) \\
\boldsymbol{\mu}^k|\boldsymbol{\Gamma}^k, \boldsymbol{\mu}^0, \boldsymbol{\beta}^0 &\sim N(\boldsymbol{\mu}^0, 1/\boldsymbol{\beta}^0(\boldsymbol{\Gamma}^k)^{-1}) \\
(\mathbf{x}, \mathbf{y})|z, \boldsymbol{\Gamma}, \boldsymbol{\mu} &\sim N(\boldsymbol{\mu}^k, (\boldsymbol{\Gamma}^k)^{-1}).
\end{aligned} \tag{3}$$

Here the superscript k denotes the cluster, and $\boldsymbol{\Gamma}^k$ is the block-diagonal precision matrix for cluster k that consists of $\boldsymbol{\Gamma}_x^k$ and $\boldsymbol{\Gamma}_y^k$, both drawn from the Wishart distribution W. The model is illustrated graphically in Figure 2, and it is very similar to the model proposed in [2] for a multivariate Gaussian mixture with non-constrained covariance matrices. The crucial difference here is the block diagonal covariance matrix which forces the model to focus on modeling dependencies.

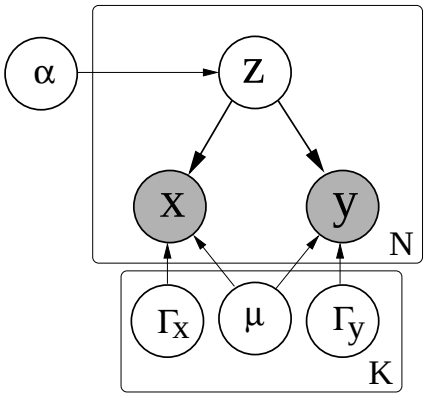

Fig. 2. Graphical representation of the Bayesian clustering model for detecting dependencies between data sets. Shaded nodes represent observed variables, whereas the rest are latent variables (z) or model parameters. The plates indicate repetition over samples (N) and clusters (K), and the hyperparameters have not been drawn for clarity.

As values for the hyperparameters we use

$$\lambda^0 = 1, \quad \beta^0 = 1, \quad \nu_x^0 = d_x, \quad \nu_y^0 = d_y$$

for the scale parameters, and the mean and precision hyperparameters are set by empirical Bayes based on the observed data. In detail, $\boldsymbol{\mu}^0$ is set to the mean of the data, and $\boldsymbol{\Psi}_x^0$ is given by $0.1\nu_x^0\boldsymbol{\Lambda}_x$ where $\boldsymbol{\Lambda}_x$ is a diagonal matrix containing the variances of $\mathbf{x}$ (and similarly for $\mathbf{y}$). This means that the prior covariance of the clusters is diagonal with variance proportional to that of the data.

Denote all model parameters collectively by Θ, and use V to denote all observed variables (both $\mathbf{x}$ and $\mathbf{y}$) and Z for all latent (unobserved) variables. In the Bayesian approach we are interested in the posterior distribution $P(\Theta, Z|V)$ (and here eventually $P(Z|V)$ if the focus is solely on detecting the shared signal), which would be the justified distribution of the possible answers given the observed data. Unfortunately solving the posterior exactly is

intractable for most interesting models, including (3). Several methods, such as sampling (see e.g. [7]), variational approximation [10] and expectation propagation [13] have been proposed for approximative inference, each having their own advantages and disadvantages. Here a variational Bayes (VB) approach is adopted.

We will here briefly describe the VB to the degree necessary for deriving a posterior approximation for (3). In VB the posterior distribution $P(\Theta, Z|V)$ is approximated by a variational distribution $Q(\Theta, Z)$, where tractability is achieved by assuming that the latent variables and parameters are independent given the observed variables, $Q(\Theta, Z) = Q(\Theta)Q(Z)$. The task is then to find the $Q(\Theta, Z)$ such that the difference between $P(\Theta, Z|V)$ and $Q(\Theta, Z)$, measured by the Kullback-Leibler divergence

$$KL(Q||P) = \iint Q(\Theta, Z) \log \frac{Q(\Theta, Z)}{P(\Theta, Z|V)} d\Theta dZ$$

is minimized. Equivalently, we can write $\log(P(V)) = L(Q) + KL(Q||P)$, where

$$L(Q) = \iint Q(\Theta, Z) \log \frac{P(V, Z, \Theta)}{Q(\Theta, Z)} d\Theta dZ, \tag{4}$$

which shows that minimizing the KL-divergence maximizes a lower bound $L(Q)$ for the true marginal likelihood $\log(P(V))$, since the KL-divergence is always positive, and zero only if $Q(\Theta, Z) = P(\Theta, Z|V)$.

We choose a fully factorized form for the approximation $Q(\Theta, Z)$. Then the form of the approximation follows directly from free-form optimization of the objective function, and we need not specify the functional form. In an iterative updating scheme we can update Z and Θ alternatingly, which directly leads to $Q(Z) \propto \exp\left(\int_\Theta Q(\Theta) \log P(D, L|\Theta) d\Theta\right)$. Furthermore, it can be shown that for exponential family distributions using conjugate priors the optimal form

for the approximation $Q(\Theta)$ is the same as that of the prior for the actual model. In the model (3) only conjugate priors are used, and thus we get

$$Q(\Theta) = Q(\boldsymbol{\alpha}) \prod_k Q(\boldsymbol{\Gamma}^k) Q(\boldsymbol{\mu}^k | \boldsymbol{\Gamma}^k),$$

where $Q(\boldsymbol{\alpha}) = \mathrm{Dir}(\boldsymbol{\lambda})$ and $Q(\boldsymbol{\mu}^k | \boldsymbol{\Gamma}^k) = N(\boldsymbol{\rho}^k, 1/\beta^k (\boldsymbol{\Gamma}^k)^{-1})$. Due to the block-diagonal nature of the covariances we have $Q(\boldsymbol{\Gamma}^k) = W(\nu_x^k, \boldsymbol{\Psi}_x^k) W(\nu_y^k, \boldsymbol{\Psi}_y^k)$.

The parameters of the approximation can be learned using an algorithm closely resembling EM. Since the model is very close to a normal Gaussian mixture model we derive the update formulas following [2]. The only difference between the models is in the parameterization of the covariance matrices, and consequently changes need to be made only to the parts involving the covariance matrix. This directly gives us the expectation formula

$$\gamma_{kn} \propto \tilde{\boldsymbol{\alpha}}^k (\tilde{\boldsymbol{\Gamma}}^k)^{1/2} e^{-(\mathbf{v}_n - \boldsymbol{\rho}^k)^T \bar{\boldsymbol{\Gamma}}^k (\mathbf{v}_n - \boldsymbol{\rho}^k)} e^{-\frac{d_v}{2\beta^k}} \tag{5}$$

where γ_{kn} denotes the probability for the nth sample to belong to the kth cluster. Here $\log \tilde{\boldsymbol{\alpha}}^k = \psi(\lambda^k) - \psi(\sum_j \lambda^j)$, $\log \tilde{\boldsymbol{\Gamma}}^k = \sum_{j=1}^{d_x} \psi((\nu_x^k + 1 - j)/2) + \sum_{j=1}^{d_y} \psi((\nu_y^k + 1 - j)/2) - \log |\boldsymbol{\Psi}_x^k| - \log |\boldsymbol{\Psi}_y^k| + d_v \log(2)$, and $\bar{\boldsymbol{\Gamma}}^k$ is a block-diagonal matrix that contains $\nu_x^k (\boldsymbol{\Psi}_x^k)^{-1}$ and $\nu_y^k (\boldsymbol{\Psi}_y^k)^{-1}$ on its diagonal. The $\psi(\cdot)$ denotes the digamma function, derivative of the logarithm of the gamma function. In all formulas d denotes the dimensionality of the corresponding variable, and $\mathbf{v}_n = [\mathbf{x}_n; \mathbf{y}_n]$ is the concatenation of the values for the nth sample.

For the maximization step we first define the following terms:

$$N^k = \sum_{n=1}^{N} \gamma_{kn}, \quad \bar{\boldsymbol{\mu}}^k = \frac{1}{N^k} \sum_{n=1}^{N} \gamma_{kn} \mathbf{v}_n,$$
$$\boldsymbol{\Sigma}^k = \frac{1}{N^k} \sum_{n=1}^{N} \gamma_{kn} (\mathbf{v}_n - \bar{\boldsymbol{\mu}}^k)(\mathbf{v}_n - \bar{\boldsymbol{\mu}}^k)^T.$$

They enable expressing the updates as

$$\lambda^k = \lambda^0 + N^k, \quad \beta^k = \beta^0 + N^k, \quad \boldsymbol{\mu}^k = (N^k \bar{\boldsymbol{\mu}}^k + \beta^0 \boldsymbol{\rho}^0)/(N^k + \beta^0), \tag{6}$$

following [2]. In practice we can re-utilize the update formulas from [2] for the covariance matrix as well, but they have to be applied to blocks corresponding to $\mathbf{x}$ and $\mathbf{y}$ separately. Each block is equivalent to the whole covariance matrix in the basic Gaussian mixture, giving

$$\nu_x^k = N^k + \nu_x^0, \quad \boldsymbol{\Psi}_x^k = N^k \left(\boldsymbol{\Sigma}_x^k + \frac{\beta^0}{N^k + \beta^0} (\bar{\boldsymbol{\mu}}_x^k - \boldsymbol{\rho}_x^0)(\boldsymbol{\mu}_x^k - \boldsymbol{\rho}_x^0)^T \right) + \boldsymbol{\Psi}_x^0 \tag{7}$$

as the update formula for the covariance block of $\mathbf{x}$ and similarly for $\mathbf{y}$. The subscript x denotes picking the part related to the $\mathbf{x}$ from the corresponding parameter that was defined for $\mathbf{v} = [\mathbf{x}; \mathbf{y}]$.

In summary, we have an iterative algorithm that proceeds by alternating two steps: (i) Estimation of the latent signal using (5), and (ii) Updating the parameters of the posterior approximation $Q(\Theta, Z)$ using (6) and (7). The algorithm is run until the lower bound $L(Q)$ in (4) stops improving, and the final value of $L(Q)$ can be used for model order selection as it is effectively the marginalized likelihood of the model minus a term that penalizes for model complexity in a justified way.

5 Experiments

Here we verify empirically some of the properties claimed in the previous sections. These experiments are not a comprehensive study on the performance of the methods, but aim to demonstrate the kinds of effects one should anticipate, and know how to deal with, when using probabilistic models for dependency

exploration tasks.

5.1 CCA and complexity of the data set-specific models

In Section 3 we claimed that the generative model only implements CCA when sufficiently complex models for the data set-specific effects are used. Here we demonstrate that the EM algorithm for extended CCA indeed converges to the classical CCA solution given full complexity, and show that this does not hold for lower complexity.

For this purpose we used a simple generated data set that has a subspace (four dimensions) with significant correlation, while the rest of the dimensions (eight) in both data spaces are independent noise. The data was sampled from a single Gaussian with the given dependency structure, the correlations being $0.9, 0.6, 0.3$, and 0.2, and three of the independent dimensions in both data sets having larger variance than the shared dimensions ($2.0, 3.0$, and 4.0 compared to the 1.0 for all the other dimensions). A set of 1000 samples was drawn from the distribution, and the solutions were computed using the EM algorithm (Table 1) for various complexities of the data set-specific models. The results were computed as averages over 100 different data sets from the same distribution.

For each model complexity we compute a four-dimensional projection (4-dimensional latent variables $\mathbf{z}$), and solve the rotational ambiguity using the procedure described in [1] to obtain four separate components. In Figure 3 the correlations extracted by these components are shown as a function of the data set-specific model complexity, and it is evident that too low a complexity

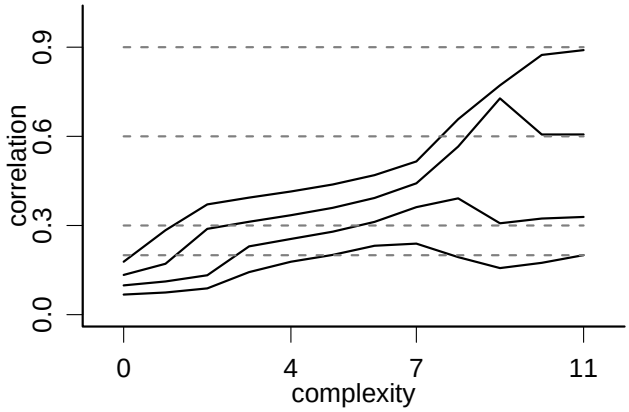

Fig. 3. Demonstration that the data set-specific models need to be of full complexity for the shared model to capture the dependencies between the data sets. On the x-axis we have the dimensionality of the latent variables of the data set-specific models, and with maximal value (here 11) we obtain the four true correlations (0.9, 0.6, 0.3 and 0.2), with slight discrepancy due to the relatively small number of samples. If less complex data set-specific models are used the shared model underestimates the correlations, and eventually with zero complexity (the leftmost point) the method reduces to probabilistic PCA that does not focus on correlations at all. The four lines depict the correlations on the four components extracted by the method, and dashed horizontal lines mark the true correlations.

leads to decreased performance in finding the canonical directions. Only the rightmost solution with full complexity finds the true correlations, and already the solution with complexity of just 2 fewer dimensions mixes severely the two largest components, while still finding roughly the correct subspace.

5.2 Relationship between CCA and the clustering model

In order to demonstrate what the clustering model does, and in particular its relation to the CCA, we run it on a simple toy data set. In Figure 4 two two-dimensional data sets are presented, with projection vectors indicating the

CCA direction. The clustering model was applied to the same data, and the cluster centroids and covariance matrices of the clusters have been overlaid on top of the figure. The number of clusters was chosen so that the lower bound for the marginalized likelihood (4) was maximized.

It is worth noting that the cluster centroids do not lie on the CCA projection, but instead follow the distribution of the data. The important thing is that when projected to the CCA component the clusters are well separated, indicating that the same structure that is found by CCA is retained. The model still tries to describe the generative process of the data, which forces the centroids to be closer to the actual data. Here the CCA solution is still fairly good, but if the data sets were further away from being Gaussian the clustering model would be a clear improvement over CCA.

5.3 Analysis of yeast stress

As a more realistic example, we cluster yeast genes based on expression measurements made in different stressful treatments. The measurement data was obtained from [6], and preprocessed as in [14]. We treat the time series measured in different conditions as data sets, and seek to cluster the genes so that dependencies between the measurements are captured by the cluster structure. The common thing between the measurements should be that all contain a general stress response, and here we can externally validate the performance of the clustering algorithm by checking whether it groups known environmental stress response (ESR) genes [6] into the same clusters. This is a two-class problem, each gene either is or is not an ESR gene. If the clustering takes the dependencies correctly into account the cluster index should correlate with

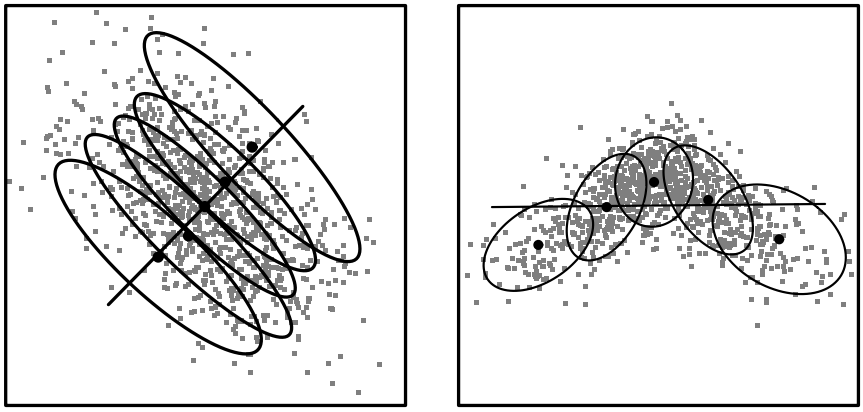

Fig. 4. Illustration of the relationship between how the clustering model and CCA. The two figures present two two-dimensional data sets that have high correlation (0.85) along a one-dimensional subspace. The CCA solution is drawn as a line to both figures, and the centroids and covariance matrices of the clustering model are also displayed. Here the clusters find essentially the same dependency as CCA, but still describe the data well by following the shape of the data. Note that in the left subfigure the variation is two times higher in the direction orthogonal to the dependent subspace, but the cluster centroids still lie on the one-dimensional CCA–subspace due to being able to model the data set-specific variation by stretching the covariance matrices.

the sample being an ESR gene.

We took a set of five different conditions with total dimensionality of 38, and compared the clustering model to the naive alternative of clustering the concatenation of the data sets using a joint distribution model, here a mixture of Gaussians with a non-constrained covariance matrix [2]. Both models use a variational approximation. As a goodness measure we use the "classification accuracy" of the clustering: for each left-out sample we calculate how big a proportion of the training samples in the same cluster belonged to the same class. The final measure is the average of that measure over all ESR genes in the left-out data, essentially measuring the capability of the clustering to collect

ESR genes into clusters with few non-ESR genes. The leave-out procedure was constructed so that a random subset of samples (half) was chosen for training and the methods were tested on the remaining half. The presented results are averaged over 10 such randomizations.

The five data sets allow studying 10 different pairings of two data sets. The results for the clustering model (3) and the comparison method have been collected into Table 2. The number of clusters for both methods was fixed to 8, which gave the best lower bound for the true marginal likelihood for the comparison model in a preliminary test with a random subsample of the data; the proposed model would have supported a larger number of clusters, but in order to not bias the goodness measure we chose a suboptimal value for our model.

We see that in 6 cases the clustering model using the block-structured covariance matrix gives significantly (p-value below 0.01; t-test) better score, indicating that the ESR labeling correlates with the clustering. In 4 of the remaining cases the difference is not significant, and in only one case the average accuracy is higher for the comparison method. The results illustrate also how the choice of the data sets is important; if there is no interesting correlation structure then focusing on the dependencies cannot help, and if the dependency is not related to the effect we are studying then it drives the results to the wrong direction.

Note also that for all pairs not involving DTT the comparison method produced a significantly better lower bound for the likelihood, revealing that if our task was simply to find the best model for the whole collection we should have chosen that model instead. This illustrates that finding the dependencies

Table 2

Accuracy of the clustering methods in detecting the ESR genes from pairs of measurements of stressful experiments. Each cell presents the results for an analysis run on a pair of experiments. The first figure is the accuracy of the dependency-seeking clustering method (in percentage units), and the second the accuracy of a standard joint model. Boldface indicates significant differences (paired t-test). A random allocation of genes into clusters would give an accuracy of 14.1%.

	Heat2	Nitrogen	Diamide	DTT
Heat1	**46.1/38.5**	**49.1/40.7**	43.3/40.0	**44.4/39.0**
Heat2	-	**50.7/45.3**	44.8/44.2	**48.1/41.8**
Nitrogen	-	-	**48.8/42.7**	47.8/45.1
Diamide	-	-	-	42.8/43.5

and describing the whole data are different tasks, even though we can use standard generative modeling techniques for both.

Even though all the formulas have been presented for two data sets the method generalizes directly to more than two. To demonstrate the performance on a larger collection we ran the same experiment for a collection of three, four and five data sets. For the collection of both heat-shocks and nitrogen depletion the average accuracy of the model (3) was 51.1%, and adding diamide treatment to the collection increased it further to 51.5%. Finally, using all five data sets gave an accuracy of 51.6%. While the differences between these three figures are not significant, they are all above the best accuracy obtained with any pair of data sets. For the comparison method the corresponding accuracies were 41.4%, 41.9% and 45.3% using the same collections. Here the best results from a pair of data sets are actually better than what was obtained using three or four

data sets, and the proposed clustering method is always significantly better than the comparison method, despite the comparison method still providing significantly better lower bound for the likelihood.

6 Discussion

In this paper we studied the use of generative models for finding dependencies between data sets. Traditionally, dependencies have been sought by explicitly optimizing a measure of dependency, using methods such as Canonical Correlation Analysis (CCA) or various clustering methods to optimize the mutual information. Recently CCA has been interpreted as a generative model, which lead us to study whether generative models could be used for dependency exploration tasks even more generally.

We specified a general model family for data fusion, including dependent (shared) and data set-specific signals. We showed that a basic probabilistic treatment leads to CCA in the case of linear projections and Gaussian noise. In particular, it was shown that the dependencies are correctly found only when the model family is such that the data set-specific part of the model is flexible enough to describe all the variation in each data set. Furthermore, the latent signals for those parts need to be marginalized, which is made tractable in the model family by the assumption of the observed signal being an additive combination of shared and data set-specific signals.

To give an example of generalizations from the basic CCA model we derived a clustering method capable of detecting dependencies, based on the same model structure. The model was treated in a fully Bayesian way by presenting

a variational Bayes approximation for the posterior distribution. The model shows how the full advantage of the extra robustness provided by the generative interpretation of dependency seeking methods can be utilized, giving for example a justified criterion for choosing the number of clusters. Recently also CCA has been treated in a fully Bayesian way [12,19], complementing the solution provided here for the clustering model.

The presented models here were reasonably simple, always making the assumption that variation within each data set can be explained by a linearly transformed Gaussian latent variable. The main reason for this was computational tractability. Even though the assumption of additive signals simplifies the marginalization process, further approximations are probably needed to develop methods where the assumptions for the data set-specific signals would be more complex. Extending to models with more complex shared signal, either in form of non-linear mapping or non-Gaussian latent variable, should be easier, since we need not marginalize over the shared latent variables.

7 Acknowledgments

This work was supported by the Academy of Finland, decision number 207467, and in part by the IST Programme of the European Community, under the PASCAL Network of Excellence, IST-2002-506778. This publication only reflects the authors' views.

References

[1] C. Archambeau, N. Delannay, M. Verleysen, Robust probabilistic projections, in: W. Cohen, A. Moore (Eds.), Proceedings of the 23rd International conference on machine learning, ACM, 2006, pp. 33–40.

[2] H. Attias, A variational Bayesian framework for graphical models, in: Advances in Neural Information Processing Systems 12, MIT Press, 2000, pp. 386–392.

[3] F. R. Bach, M. I. Jordan, A probabilistic interpretation of canonical correlation analysis, Tech. Rep. 688, Department of Statistics, University of California, Berkeley (2005).

[4] S. Becker, G. E. Hinton, Self-organizing neural network that discovers surfaces in random-dot stereograms, Nature 355 (1992) 161–163.

[5] I. S. Dhillon, S. Mallela, D. S. Modha, Information-theoretic co-clustering, in: Proceedings of KDD'03, The Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM Press, New York, NY, USA, 2003, pp. 89–98.

[6] A. P. Gasch, P. T. Spellman, C. M. Kao, O. Carmel-Harel, M. B. Eisen, G. Storz, D. Botstein, P. O. Brown, Genomic expression programs in the response of yeast cells to environmental changes, Molecular Biology of the Cell 11 (2000) 4241–4257.

[7] A. Gelman, J. B. Carlin, H. S. Stern, D. B. Rubin, Bayesian Data Analysis (2nd edition), Chapman & Hall/CRC, Boca Raton, FL, 2003.

[8] D. R. Hardoon, S. Szedmak, J. Shawe-Taylor, Canonical correlation analysis: An overview with application to learning methods, Neural Computation 16 (12) (2004) 2639–2664.

[9] H. Hotelling, Relations between two sets of variates, Biometrika 28 (1936) 321–377.

[10] M. I. Jordan, Z. Ghahramani, T. Jaakkola, L. Saul, An introduction to variational methods for graphical models, in: M. I. Jordan (Ed.), Learning in Graphical Models, MIT Press, Cambridge, 1999, pp. 105–162.

[11] S. Kaski, J. Nikkilä, J. Sinkkonen, L. Lahti, J. Knuuttila, C. Roos, Associative clustering for exploring dependencies between functional genomics data sets, IEEE/ACM Transactions on Computational Biology and Bioinformatics, Special Issue on Machine Learning for Bioinformatics – Part 2 2 (3) (2005) 203–216.

[12] A. Klami, S. Kaski, Local dependent components, in: Z. Ghahramani (Ed.), Proceedings of ICML 2007, the 24th International Conference on Machine Learning, Omnipress, 2007, pp. 425–432.

[13] T. Minka, Expectation Propagation for approximative Bayesian inference, in: J. S. Breese, D. Koller (Eds.), Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence, 2001, pp. 362–369.

[14] J. Nikkilä, C. Roos, E. Savia, S. Kaski, Explorative modeling of yeast stress response and its regulation with gCCA and associative clustering, International Journal of Neural Systems 15 (4) (2005) 237–246.

[15] S. Roweis, Z. Ghahramani, A unifying review of linear gaussian models, Neural Computation 11 (1999) 305–345.

[16] J. Sinkkonen, S. Kaski, Clustering based on conditional distributions in an auxiliary space, Neural Computation 14 (2002) 217–239.

[17] M. Tipping, C. Bishop, Mixtures of probabilistic principal component analysers, Neural Computation 11 (1999) 443–482.

[18] N. Tishby, F. C. Pereira, W. Bialek, The information bottleneck method, in: B. Hajek, R. S. Sreenivas (Eds.), Proceedings of The 37th Annual Allerton Conference on Communication, Control, and Computing, University of Illinois, Urbana, Illinois, 1999, pp. 368–377.

[19] C. Wang, Variational Bayesian approach to canonical correlation analysis, IEEE Transactions on Neural Networks 18 (2007) 905–910.