Abstract
Canonical correlation analysis (CCA) finds correlating components of two multivariate random variables. Recently it was shown how CCA can be computed by incorporating a specific kind of group sparsity assumption into a standard PCA model for the concatenation of the two random variables, resulting in a CCA variant that is computationally more efficient and applicable for high-dimensional data. Interestingly, the same approach generalizes to multiple data sources instead of just two, resulting in a new multiset CCA generalization that is more flexible than the existing ones. The remaining open question is how to most efficiently learn the sparse structure.

1. Introduction
Canonical correlation analysis (CCA) finds linear projections of two random variables $y_1 \in \mathbb{R}^{D_1}$ and $y_2 \in \mathbb{R}^{D_2}$ so that the projections $w_1^T y_1$ and $w_2^T y_2$ correlate maximally. CCA can be used, for example, for integrating two co-occurring data sources, called views, into a single representation capturing the correlating components. In addition, the view-specific components independent of the other view may be useful.

CCA as such is defined only for two variables or views, largely since correlation itself is a bivariate construct. Over the years multiple generalizations of CCA to multiple views have been presented, based on the ideas of Kettenring (1971). Some recent works call these methods multiset CCA and apply them e.g. for finding functional connectivity from fMRI data (Deleus & Van Hulle, 2011). Typically the methods are based on the same generalization best interpreted as seeking components that are active in all views. For example, Tripathi et al. (2008) whitens all data sources and then seeks for maximal variance in the concatenation of the sources, whereas the probabilistic multiset generalization by Archambeau & Bach (2009) assumes shared latent signals visible in each view.

An interesting alternative generalization would allow not only components shared by all views but also components shared by any subset of the views, including the singletons providing view-specific components. Unfortunately, there are exponentially many such subsets, preventing efficient models with explicit components of all types. In this work we show how such a generalization can still be constructed, by utilization of group sparsity. Recently Virtanen et al. (2011) showed how the solution of the standard two-view CCA can, surprisingly, be computed with a PCA model incorporating group sparsity. Extending that approach into more than two views gives a multiset CCA model that allows components shared by any possible subset of the views while still being computationally efficient.

The method relies on group sparsity to select the subsets that are necessary for modeling the observed data. Consequently, the performance of the method heavily depends on how well the correct structure is found. In this work we apply automatic relevance determination (ARD) in a group-wise manner and boost the convergence with an improved variational approximation explicitly maximizing independence of the components, but expect that alternative techniques for finding group-wise sparse solutions would be useful as well.

2. Method
Following Archambeau & Bach (2009); Klami & Kaski (2008); Virtanen et al. (2011), the Bayesian CCA
model is given by

\[
\begin{align*}
  z &\sim \mathcal{N}(0, I) \\
  x &\sim \mathcal{N}(Wz, \Psi)
\end{align*}
\]  

where \( z \in \mathbb{R}^K \) is a latent variable, \( W \in \mathbb{R}^{D_1+D_2 \times K} \) is a projection matrix with a specific structure explained below, and \( x = [y_1; y_2] \) is the concatenation of the two multivariate random variables. Furthermore, \( \Psi \in \mathbb{R}^{D_1+D_2 \times D_1+D_2} \) is a diagonal noise covariance with equal value \( \sigma_1^2 \) for the first \( D_1 \) elements and \( \sigma_2^2 \) for the last \( D_2 \) elements.

The model that looks on surface like Bayesian PCA or factor analysis becomes CCA when the projection is constrained to follow the block-wise structure of

\[
W = \begin{bmatrix} W_1 & V_1 & 0 \\ W_2 & 0 & V_2 \end{bmatrix}
\]  

Here \( W_i \) capture the correlations and \( V_i \) contain the view-specific components. The number of columns for each sub-matrix is \textit{a priori} unknown, so that the total number equals the latent dimensionality \( K \).

An efficient way of learning the model is to use a PCA algorithm with added group sparsity prior for learning the correct structure for \( W \). The variables corresponding to the two views are interpreted as two groups, and solutions pushing all of the projection weights corresponding to either group towards zero are favored, separately for each component. Re-arranging the columns then always gives a structure like (2). Components with zero values for both groups are discarded, resulting in automatic complexity control.

The generalization to multiple views is straightforward. We still use a PCA model with group sparsity, but now have \( M \) groups of variables for the \( M \) views or data sources. For each component and each group the model either pushes the projection coefficients to zero or not, resulting in components that are active in some subset of views. The model then learns structures like in Fig. 1 (left), instead of the more restricted ones of earlier multiset CCA models (Fig. 1 (right)).

3. Discussion

The interesting connection between group-wise sparse PCA and (multiset) CCA holds irrespective of the computational approach. Virtanen et al. (2011), uses group-wise ARD prior. The structure is learned by a variational approximation maximizing the independence of the latent components by optimizing the variational lower bound with respect to an affine transformation of the components. In our experiments this approach has given good performance, but more advanced solutions exploring the correct structure amongst the exponentially many combinations could help for complex real-world data sets.

Acknowledgments. We thank Academy of Finland (dec.no. 133818 and AIRC Center of Excellence), the aivoAALTO project, and PASCAL2 for support.

References


