Probabilistic dependency modeling toolkit

Leo Lahti*and Olli-Pekka Huovilainen
Department of Information and Computer Science
Aalto University, Finland

February 14, 2011

1 Introduction

This package provides general tools for the discovery and analysis of statistical dependencies between co-occurring measurement data [6]. The tools include well-established models such as probabilistic canonical correlation analysis [2]. Probabilistic framework deals rigorously with the uncertainties associated with small sample sizes, and allows incorporation of prior information in the analysis through Bayesian priors [5]. The applicability of the models has been demonstrated in previous case studies [4, 9].

Dependency models help to discover regularities and interactions that are not seen in individual data sets. Multiple, complementary views of the same objects are available in many fields including computational biology, economics, linguistics, neuroinformatics, open data initiatives, social sciences, and other domains. Demand for methods that can reveal dependencies between heterogeneous observations is increasing with the availability of co-occurring observations. Open access implementations of the algorithmic solutions help to realize the full potential of these information sources.

Your feedback and contributions are welcome.[1]

1.1 Installation

Install dmt from within R using command
'install.packages("dmt", repos="http://R-Forge.R-project.org")'

2 Regularized dependency detection

The fit.dependency.model function is used to detect dependencies between two data sets, X and Y:

```
> library(dmt)
```

*leo.lahti@iki.fi
[1]See the project page at R-Forge: http://dmt.r-forge.r-project.org/

```
dmt Copyright (C) 2008-2011 Leo Lahti and Olli-Pekka
Huovilainen.
This program comes with ABSOLUTELY NO WARRANTY.
This is
free software, and you are welcome to redistribute it under FreeBSD, see the licensing ter
```

```
> data(modelData)
> model <- fit.dependency.model(X, Y)
```

The underlying linear model is described in Section˜3.

3 Probabilistic dependency modeling framework

The *fit.dependency.model* function implements the probabilistic dependency modeling framework presented in [2] and its subsequent extensions [1, 3, 4]. The latent variable model assumes that the two data sets, X and Y can be decomposed in *shared* and *data set-specific* components (Figure˜1). We provide tools to discover these components.

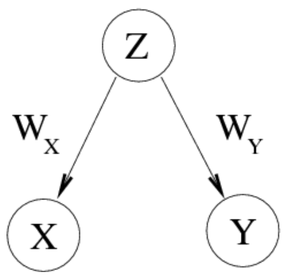

Figure 1: Graphical description of the shared latent variable model. The model assumes a shared latent variable $\mathbf{z}$, with dataset-specific manifestations ($W_x\mathbf{z}$ and $W_y\mathbf{z}$). The observed data sets, X and Y, consist of shared and dataset-specific components. The model assumes co-occurring observations, i.e. paired samples in the two data sets.

The shared signal is modeled with a latent variable $\mathbf{z}$. Intuitively, this measures the strength of the shared signal in each sample. The shared signal can have different manifestation in each data set, as described by the linear transformations W_x and W_y.

A standard Gaussian model for the shared latent variable $\mathbf{z} \sim N(0, I)$ and data set-specific effects gives the following model where the data set-specific effects are modelled by the covariance matrices Ψ_x, Ψ_y.

$$\begin{aligned} X &\sim W_x\mathbf{z} + \varepsilon_x \\ Y &\sim W_y\mathbf{z} + \varepsilon_y \\ \varepsilon_. &\sim \mathcal{N}(0, \Psi_.) \\ \mathbf{z} &\sim \mathcal{N}(0, I) \end{aligned} \tag{1}$$

The options of the fit.dependency.model function provides can be used to tune the model structure. For instance, it is possible to tune the dimensionality of the latent variable and to regularize model parameters. An overview is provided below.

3.1 Regularized dependency detection

Various options are available to tune the model structure and to guide dependency modeling through Bayesian priors in the 'priors' option in the fit.dependency.model function [4]. For instance, the following will fit a model with $W_x = W_y$, with non-negative (but otherwise unconstrained) W_x, W_y and with full marginal covariances for the dataset-specific effects.

```
> model <- fit.dependency.model(X, Y, priors = list(Nm.wx.wy.sigma = 0,
+     Nm.wx.wy.mean = 1, W = 0.001), marginalCovariances = "full")
```

Below is a brief summary of the available options. For further options and examples, see help(fit.dependency.model):

zDimension	Dimensionality of the latent variable, which is used to characterize the latent effects. By default, full dimensionality is used, but in many applications the relevant dependencies can be described with lower-dimensional representation of the shared effects.
W	By default, no constraints are applied on W_x and W_y. However, non-negative solutions can be obtained by setting an exponential prior with rate parameter W: $W_x \sim exp(-WW_x(i))$ for each element i of the matrix W_x, and respectively for W_y. Small values of the rate parameter enforce non-negativity but are otherwise non-informative.
marginalCovariances	Dataset-specific effects can come from Gaussian distribution with either full, diagonal, isotropic, or identical isotropic covariance structure. The last option refers to a model with $\Phi_x = \Phi_y$.
matched	If matched = TRUE, it is possible to tune the relationship between W_x and W_y. See Nm.wx.wy.sigma.
m.wx.wy.sigma, Nm.wx.wy.mean	Assuming that $W_y = TW_x$, it is possible to tune the relationship between W_x and W_y through a prior on T. This can be useful for guiding the modeling to focus on certain types of dependencies, and to avoid overfitting. Here, a matrix normal distribution is applied: $T \sim N_m(H, \sigma * I, \sigma I)$, with mean and covariance $H = Nm.wx.wy.mean$, $\sigma I = Nm.wx.wy.sigmaI$, respectively. By default, Nm.wx.wy.mean = I. The prior can be tuned through σ. When $\sigma = 0$, $W_x = W_y$; when $\sigma \to \infty$, the relationship between W_x and W_y is not constrained.

3.2 Special cases

Special cases of the model include probabilistic versions of canonical correlation analysis, factor analysis, and principal component analysis, and their regularized variants.

Probabilistic CCA (pCCA) assumes full covariance matrices Ψ_x, Ψ_y. This gives the most detailed model for the data set specific effects. The connection of this latent variable model and the traditional canonical correlation analysis has been established in [2].

Probabilistic SimCCA (pSimCCA) assumes full covariance matrices Ψ_x, Ψ_y and identical latent transformations $W_x = W_y$. The model is more robust to overfitting than pCCA when the data is scarce [4].

Probabilistic factor analysis (pFA) is obtained with diagonal covariances Ψ_x, Ψ_y. In addition, a special case is implemented where each covariance matrix $\Psi_.$ is isotropic but not necessarily identical (as would be the case in pPCA). This model is identical to concatenating X, Y, and fitting ordinary probabilistic factor analysis on the concatenated data set. The structure of the covariances is simpler than in pCCA. This regularizes the solution and can potentially reduce overfitting in some applications. Also the standard probabilistic factor analysis model for a single data set is available [7].

Probabilistic PCA (pPCA) is obtained with identical isotropic covariances for the data set-specific effects: $\Psi_x = \Psi_y = \sigma I$. This model is identical to concatenating X, Y, and fitting ordinary probabilistic PCA on the concatenated data. Also the standard probabilistic PCA model for a single data set is available [8].

3.3 Parameter estimation

Model parameters are estimated with an EM algorithm. Conventional optimization methods are used when no analytical solution exists for particular variables.

4 Dependency-based dimensionality reduction (drCCA)

The drCCA algorithm [9] provides tools for dimensionality reduction and data fusion that retains the variation shared between the original data sources, while reducing data set-specific effects based on a linear projections. The algorithms combine data sets with co-occurring samples into a common representation of low dimensionality based on linear transformations through generalized CCA. This helps to discover dependencies between multiple data sets (two or more) simultaneously. Regularization options and automated tools are available to select the final dimensionality of the combined data set. This example shows how to perform dependency-based dimension reduction on two data sets (samples x features matrices).

```
> data(expdata1)
> data(expdata2)
> drcca <- drCCAcombine(list(expdata1, expdata2))
```

```
[1] "Number of input matrices"
[1] 2
[1] "Number of test and training matrices created"
[1] 3
[1] "Normal gCCA"
[1] 2000   12
```

```
> r <- regCCA(list(expdata1, expdata2))
```

```
[1] "Normal gCCA"

> shared <- sharedVar(list(expdata1, expdata2), r, 4)
```

Linear projections are identified for each individual data set, and a combined representation of the dependent components is constructed on a lower-dimensional space. See the original publication for further details [9]:

5 Details

- *Licensing terms:* the package is licensed under FreeBSD open software license.
- *Citing DMT:* Please cite [6, 5] when using the package; for particular algorithms (drCCA, fit.dependency.model, etc.), see separate citation information in the function help.

This document was written using:

```
> sessionInfo()

R version 2.12.1 (2010-12-16)
Platform: x86_64-unknown-linux-gnu (64-bit)

locale:
 [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
 [3] LC_TIME=en_US.UTF-8        LC_COLLATE=C
 [5] LC_MONETARY=C              LC_MESSAGES=en_US.UTF-8
 [7] LC_PAPER=en_US.UTF-8       LC_NAME=C
 [9] LC_ADDRESS=C               LC_TELEPHONE=C
[11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base

other attached packages:
[1] dmt_0.7.02         ellipse_0.3-5      Matrix_0.999375-46 lattice_0.19-13
[5] MASS_7.3-9         mvtnorm_0.9-96

loaded via a namespace (and not attached):
[1] grid_2.12.1  tools_2.12.1
```

Acknowledgements

The project is a joint effort by several people. In addition to package authors Leo Lahti and Olli-Pekka Huovilainen, particular thanks go to contributors Arto Klami and Abhishek Tripathi regarding the drCCA functionality.

References

[1] C. Archambeau, N, Delannay, and M. Verleysen. Robust probabilistic projections. In W.W. Cohen and A.˜Moore, editors, *Proceedings of the 23rd International conference on machine learning*, pages 33–40. ACM, 2006.

[2] F.R. Bach and M.I. Jordan. A probabilistic interpretation of canonical correlation analysis. Technical Report 688, Department of Statistics, University of California, Berkeley, 2005.

[3] A. Klami and S. Kaski. Probabilistic approach to detecting dependencies between data sets. *Neurocomputing*, 72(1-3):39–46, 2008.

[4] L. Lahti, S. Myllykangas, S. Knuutila, and S. Kaski. Dependency detection with similarity constraints. In *Proc. MLSP'09 IEEE International Workshop on Machine Learning for Signal Processing*, IEEE, Piscataway, NJ, 2009.

[5] L. Lahti (2010). Probabilistic analysis of the human transcriptome with side information. PhD thesis. Aalto University School of Science and Technology, Department of information and Computer Science, Espoo, Finland, 2010. http://lib.tkk.fi/Diss/2010/isbn9789526033686/

[6] L. Lahti *et al.* (2010). Dependency modeling toolkit. International Conference on Machine Learning (ICML-2010). Workshop on Machine Learning Open Source Software. Haifa, Israel, 2010. Project url: http://dmt.r-forge.r-project.org

[7] D.B. Rubin and D.T. Thayer (1982). EM algorithms for ML factor analysis. Psychometrika 47(1):69–76. url: http://www.springerlink.com/content/f7x37l8656877311/

[8] M,E. Tipping and C.M. Bishop (1999). Probabilistic principal component analysis. Journal of Royal Statistical Society B 61(3):611–622.

[9] A. Klami, A. Tripathi and S. Kaski. Simple integrative preprocessing preserves what is shared in data sources. *BMC Bioinformatics*, 9(111), 2008.