Dependency modeling toolkit

Leo Lahti*, Olli-Pekka Huovilainen, and Abhishek Tripathi Department of Information and
Aalto University, Finland

February 5, 2011

1 Introduction

Dependency modeling between multiple data sources allows the discovery of regularities and interactions that are not seen in individual data sets. The need for such methods is increasing with the availability of co-occurring observations that provide complementary views of the objects of interest in computational biology, natural language modeling, neuroinformatics, open data initiatives, social sciences, and in other domains. Open access implementations of the algorithmic solutions will help to realize the full potential of these information sources.

This package provides general-purpose tools for the discovery and analysis of statistical dependencies between co-occurring measurement data. The implementations are based on well-established models such as probabilistic canonical correlation analysis [1, 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 [3]. The applicability of the models has been demonstrated in previous case studies [3, 6]. 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 Examples

To learn dependency models for two data sets, X and Y, use the fit.dependency.model function:

```
> library(dmt)

dmt Copyright (C) 2008-2011 Leo Lahti, Olli-Pekka Huovilainen, and Abhishek Tripathi.
This program comes with ABSOLUTELY NO WARRANTY.
This is free software, and you are welcome to redistribute it under GNU GPL 2, see the lic
```

*leo.lahti@iki.fi

[1] See the project page at R-Forge: http://dmt.r-forge.r-project.org/

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

Centering the data..
[1] "wcost 0"
[1] "wcost 0"
[1] "wcost 0"
[1] "wcost 0"
[1] "wcost 0"
[1] "wcost 0"
[1] "wcost 0"
[1] "wcost 0"
[1] "wcost 0"
[1] "wcost 0"
[1] "wcost 0"
[1] "wcost 0"
[1] "wcost 0"
[1] "wcost 0"
[1] "wcost 0"
[1] "wcost 0"
[1] "wcost 0"
[1] "wcost 0"
[1] "wcost 0"
[1] "wcost 0"
[1] "wcost 0"
[1] "wcost 0"
[1] "wcost 0"
[1] "wcost 0"
[1] "wcost 0"
[1] "wcost 0"
[1] "wcost 0"
```

The functions provides additional options to tune the dimensionality of the latent variable and to regularize model parameters. An overview of the implemented probabilistic models for dependency detection is provided below. For further options, see help(fit.dependency.model).

3 Functionality

- regularized dependency detection [2, 3]
- dependency-based dimensionality reduction [6]

3.1 Documentation

The package implements the dependency modeling framework explained below (see function 'fit.dependency.model'), and provides wrappers for the special cases of the model.

4 Probabilistic dependency modeling framework

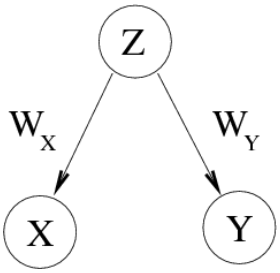

Figure 1: Graphical description of the shared latent variable model showing generation of data sets X and Y from latent shared variable $\mathbf{z}$ through W_x and W_y

The package [3] implements the probabilistic dependency modeling framework presented in [2] and extensions [1, 7, 3]. 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). Our tools help to discover these components, given modeling assumptions.

The shared signal is modeled with a latent variable $\mathbf{z}$. Intuitively, this measures the strength of the shared signal in each patient. Shared signal can have different manifestation in each data set, described by $W_x z$ and $W_y z$ where W_x, W_y. Assuming a standard Gaussian model for the shared latent variable $\mathbf{z} \sim N(0, I)$ and data set-specific effects, this leads to the following model:

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

The data set-specific effects are modelled by the covariance matrices Ψ_x, Ψ_y. Model parameters are estimated with an EM algorithm.

4.1 Special cases

Special cases of the model include probabilistic versions of canonical correlation analysis, factor analysis, and principal component analysis, and 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 factor analysis (pFA) is obtained with diagonal covariances Ψ_x, Ψ_y. In addition, a special case is implemented where each covariance matrix $\Psi_\cdot$ 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.

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.

4.2 Regularized dependency modelling

We provide toos to guide dependency modeling through Bayesian priors [3]. Prior on the relation between W_x and W_y can be used to guide modeling to focus on certain types of dependencies, and to avoid overfitting. The relationship is described through $W_y = TW_x$. We use matrix normal prior distribution: $P(T) = N_m(H, \sigma_T^2 I, \sigma_T^2 I)$. By default, $H = I$ and $\sigma_T^2 = 0$, giving $W_y = W_x$. This model is denoted pSimCCA in the package. The prior can be loosened by tuning $sigma_T^2$. With $sigma_T^2 \to \infty$, estimation of W_x and W_y become independent, yielding ordinary probabilistic CCA. It is also possible to tune the mean matrix H. This would set a particular relationship between the manifestations of the shared component in each data set, and $sigma_T^2$ is again be used to tune the strength of such prior.

5 Dependency-based dimensionality reduction

The drCCA [6] method can be used for dependency-based dimensionality reduction that retains the variation shared between the original data sources, while reducing data set-specific effects. The approach utilizes generalized canonical correlation analysis to perform a linear projection on the collection of data sets. Linearity makes it fast on large data sets. The package includes regularization and tools to select the final dimensionality of the combined data set automatically. More examples will be added later.

6 Details

- *Licensing terms:* the package is licensed under FreeBSD open software license
- *Citing DMT:* Please cite [5, 4] when using the package

This document was written using:

```
> sessionInfo()

R version 2.12.0 (2010-10-15)
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_0.4.12         ellipse_0.3-5      Matrix_0.999375-46 lattice_0.19-13
[5] MASS_7.3-8         mvtnorm_0.9-95

loaded via a namespace (and not attached):
[1] grid_2.12.0  tools_2.12.0
```


Acknowledgements

The project is a joint effort by several people: Leo Lahti, Olli-Pekka Huovilainen, and Abhishek Tripathi from the Statistical Machine Learning and Bioinformatics group at the Department of Information and Computer Science, Aalto University School of Science and Technology, Finland. Abhishek Tripathi is with Department of Computer Science, University of Helsinki, Finland. The authors belong also to Helsinki Institute for Information Technology HIIT and Adaptive Informatics Research Centre AIRC.

References

[1] Cédric Archambeau, Nicolas Delannay, and Michel 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] Francis˜R. Bach and Michael˜I. Jordan. A probabilistic interpretation of canonical correlation analysis. Technical Report 688, Department of Statistics, University of California, Berkeley, 2005.

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

[4] Leo 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/

[5] Leo 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

[6] Arto˜Klami Abhishek˜Tripathi and Samuel Kaski. Simple integrative preprocessing preserves what is shared in data sources. *BMC Bioinformatics*, 9(111), 2008.

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