

# Package ‘dmt’

January 27, 2015

**Type** Package

**Title** Dependency Modeling Toolkit

**Version** 0.8.20

**Date** 2013-12-10

**Author** Leo Lahti and Olli-Pekka Huovilainen

**Maintainer** Leo Lahti <leo.lahti@iki.fi>

**Description** Probabilistic dependency modeling toolkit.

**License** BSD\_2\_clause + file LICENSE

**Collate** AllClasses.R AllGenerics.R DependencyModel-accessors.R  
dependency.score.R fit.dependency.model.R toydata.R internals.R  
show-methods.R utilities.R set.M.R set.beta.R get.W.R  
firstlib.R drcca-functions.R optimize.parameters.R update.phi.R  
update.W.R z.expectation.R

**LazyLoad** yes

**Depends** R (>= 2.15.0), methods, mvtnorm, MASS

**Imports** Matrix

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2013-12-12 01:20:16

## R topics documented:

centerData . . . . .	2
compare.estimate.and.truth . . . . .	3
dependency.score . . . . .	4
DependencyModel-class . . . . .	5
drCCAcombine . . . . .	6
expdata . . . . .	7
fit.dependency.model . . . . .	8
generate.toydata . . . . .	11
modelData . . . . .	12

pfa.neg.log.likelihood . . . . .	13
plotVar . . . . .	14
regCCA . . . . .	15
sharedVar . . . . .	17
specificVar . . . . .	19
z.expectation . . . . .	20

<b>Index</b>	<b>22</b>
--------------	-----------

---

centerData	<i>Center data matrix.</i>
------------	----------------------------

---

### Description

Center data matrix to 0 for each variable by removing the means.

### Usage

```
centerData(X, rm.na = TRUE, meanvalue = NULL)
```

### Arguments

X	The data set: samples x features. Each feature will be centered.
rm.na	Ignore NAs.
meanvalue	Can be used to set a desired center value. The default is 0.

### Value

Centered data matrix.

### Note

Note that the model assumes samples x features matrix, and centers each feature.

### Author(s)

Leo Lahti <leo.lahti@iki.fi>

### References

See citation("dmt").

### Examples

```
#centerData(X)
```

---

`compare.estimate.and.truth`*Measuring model accuracy.*

---

**Description**

Compare estimated parameters and the original parameters used to generate the toydata.

**Usage**

```
compare.estimate.and.truth(res, toy)
```

**Arguments**

<code>res</code>	DependencyModel object.
<code>toy</code>	Toydata which was used to learn the model.

**Details**

Given original data (including the original parameters) and model learned based on the data, compare the learned parameters to the original ones.

**Value**

Vector with following elements

<code>wtw.x</code> , <code>wtw.y</code>	Correlation between the original and estimated values for $W^*t(W)$ for X and Y, respectively.
<code>phi.x</code> , <code>phi.y</code>	Correlation between the original and estimated values for phi for X and Y, respectively.

**Note**

Additional tests added later.

**Author(s)**

Leo Lahti <leo.lahti@iki.fi>

**References**

See citation("dmt") for references.

**See Also**

[generate.toydata](#)

**Examples**

```
# toy <- generate.toydata(N = N,  
#       zDim = zdim, xDim = xdim, yDim = ydim,  
#       marginal.covariances = marginalCovariances,  
#       priors = priors)  
# res <- fit.dependency.model(toy$X, toy$Y,  
#       zDimension = zdim,  
#       marginalCovariances = marginalCovariances,  
#       priors = priors,  
#       matched = FALSE)  
# vec <- compare.estimate.and.truth(res, toy)  
#
```

---

dependency.score	<i>dependency.score</i>
------------------	-------------------------

---

**Description**

Dependency score

**Usage**

```
dependency.score(model)
```

**Arguments**

model            DependencyModel object

**Value**

Dependency score

**Author(s)**

Leo Lahti <leo.lahti@iki.fi>

**References**

See citation("dmt").

**Examples**

```
#dependency.score(model)
```

---

DependencyModel-class *Class "DependencyModel"*

---

### Description

A Dependency model for one or two data sets

### Objects from the Class

Returned by `fit.dependency.model`, `ppca`, `pfa`, and `pcca` functions.

### Slots

**W** a list of X, Y and total components containing the relationship between two data sets; for dependency model for one dataset, only total is given

**phi** a list of X, Y and total components containing the data set specific covariances; for dependency model for one dataset, only total is given

**score** score for fitness of model

**method** name of the used method

**params** list of parameters used in dependency model

**data** The data used to calculate the dependency model

**z** The latent variable Z

### Methods

**getW** signature(model = "DependencyModel"): Returns a list of model variable Ws X, Y and total component

**getPhi** signature(model = "DependencyModel"): Returns a list of model variable phis X and Y and total component

**getScore** signature(model = "DependencyModel"): Returns the dependency score of model

**getParams** signature(model = "DependencyModel"): Returns a list of used parameters for the method

**getModelMethod** signature(model = "DependencyModel"): Returns the name of the used method

**getWindowSize** signature(model = "DependencyModel"): Returns the size of window

**getZ** signature(model = "DependencyModel", X = "numeric", Y = "numeric"): Returns the latent variable z. Arguments X and Y are needed only when the dependency model is calculated without calculating the latent variable and the original data is not included with the model (arguments calculateZ = FALSE and includeData = FALSE in `fit.dependency.model`).

### Author(s)

Olli-Pekka Huovilainen <ohuovila@gmail.com>

## Examples

```
data(modelData) # Load example data X, Y
model <- fit.dependency.model(X, Y)
# Getting the latent variable Z when it has been calculated with the model
#getZ(model)
```

---

drCCAcombine

*A function to combine several data sets*


---

## Description

Performs drCCA on a collection of data sets with co-occurring samples. The method utilizes regularized canonical correlation analysis to find linear projections for each of the data sets, and uses those to construct a combined representation of lower dimensionality than the original collection. The method suggests a specific dimensionality for the combined representation, but it is possible to obtain also combined data sets of different dimensionality.

## Usage

```
drCCAcombine(datasets, reg=0, nfold=3, nrand=50)
```

## Arguments

datasets	A list containing the data matrices to be combined. Each matrix needs to have the same number of rows (samples), but the number of columns (features) can differ. Each row needs to correspond to the same sample in every matrix.
reg	Regularization parameter for the whitening step used to remove data-set specific variation. The value of parameter must be between 0 and 1. The default value is set to 0, which means no regularization will be used. If a non-zero value is given it means that some of the dimensions with the lowest variance are ignored when whitening. In more technical terms, the dimensions whose total contribution to the sum of eigenvalues of the covariance matrix of each data set is below reg will not be used for the whitening.
nfold	The number of cross-validation folds used in the automatic dimensionality estimation process. The default value is 3.
nrand	The number of random comparison data-sets created for the automatic dimensionality estimation process. The default value is 50.

## Details

The function uses [regCCA](#) to perform the canonical correlation analysis. The dimensionality of the combined data set is selected using a statistical test that aims to find which dimensions capture shared variation significantly more than what would be found under the assumption that the data sets were independent. For this purpose nrand collections of random matrices with similar variance structure but no between-data dependencies are created. The amount of variation each dimension extracts from leave-out data in the cross-validation setting with nfold folds is compared to the

distribution obtained from the random matrices, and the dimensions that differ significantly from the null hypothesis of independence are kept in the combined representation. For details, please check the reference.

**Value**

The function returns a list of two values.

proj	The representation obtained by combining the source data sets. This is a matrix that contains a feature representation for each of the samples in the analyzed collection. Each row in this result matches the corresponding row in the original data sets.
n	The number of dimensions in the combined representation. This is equal to <code>ncol(proj)</code> .

**Author(s)**

Abhishek Tripathi <abhishektripathi.at@gmail.com>, Arto Klami

**References**

Tripathi A., Klami A., Kaski S. (2007), Simple integrative preprocessing preserves what is shared in data sources.

**See Also**

[regCCA](#)

**Examples**

```
# data(expdata1)
# data(expdata2)
# drCCAcombine(list(expdata1,expdata2),0,2,3)
```

---

expdata

*Example data for drCCA*

---

**Description**

Randomly generated data set for the examples. Data matrix with 2000 rows and 6 columns. Rows corresponds to the samples and each column corresponds to a particular features.

**Usage**

```
data(expdata1)
```

**Format**

A data matrix with 2000 rows and 6 columns. Column names and row names are omitted.

**Details**

A randomly generated data set for the example purpose only.

**Examples**

```
#data(expdata1)
```

---

```
fit.dependency.model  Fit dependency model between two data sets.
```

---

**Description**

Fit generative latent variable model (see vignette for model specification) on two data sets. Regularize the solutions with priors, including constraints on marginal covariance structures, the structure of  $W$ , latent dimensionality etc. Probabilistic versions of PCA, factor analysis and CCA are available as special cases.

**Usage**

```
fit.dependency.model(X, Y, zDimension = 1, marginalCovariances = "full",
                    epsilon = 1e-3,
                    priors = list(), matched = TRUE,
                    includeData = TRUE, calculateZ = TRUE, verbose = FALSE)
ppca(X, Y = NULL, zDimension = NULL, includeData = TRUE, calculateZ = TRUE)
pfa(X, Y = NULL, zDimension = NULL, includeData = TRUE, calculateZ = TRUE, priors = NULL)
pcca(X, Y, zDimension = NULL, includeData = TRUE, calculateZ = TRUE)
```

**Arguments**

<code>X, Y</code>	Data set/s X and Y. 'Variables x samples'. The second data set (Y) is optional.
<code>zDimension</code>	Dimensionality of the shared latent variable.
<code>marginalCovariances</code>	Structure of marginal covariances, assuming multivariate Gaussian distributions for the dataset-specific effects. Options: "identical isotropic", "isotropic", "diagonal" and "full". The difference between isotropic and identical isotropic options is that in isotropic model, $\phi^X \neq \phi^Y$ in general, whereas with isotropic model $\phi^X = \phi^Y$ .
<code>epsilon</code>	Convergence limit.
<code>priors</code>	Prior parameters for the model. A list, which can contain some of the following elements:



	<b>W</b> Rate parameter for exponential distribution (should be positive). Used to specify the prior for $W_x$ and $W_y$ in the dependency model. The exponential prior is used to produce non-negative solutions for $W$ ; small values of the rate parameter correspond to an uninformative prior distribution.
	<b>Nm.wxwy.mean</b> Mean of the matrix normal prior distribution for the transformation matrix $T$ . Must be a matrix of size (variables in first data set) $x$ (variables in second data set). If value is 1, <code>Nm.wxwy.mean</code> will be made identity matrix of appropriate size.
	<b>Nm.wxwy.sigma</b> Variance parameter for the matrix normal prior distribution of the transformation matrix $T$ . Described the allowed deviation scale of the transformation matrix $T$ from the mean matrix <code>Nm.wxwy.mean</code> .
matched	Logical indicating if the variables (dimensions) are matched between $X$ and $Y$ . Applicable only when $\dim X = \dim Y$ . Affects the results only when prior on the relationship $W_x \sim W_y$ is set, i.e. when <code>priors\$Nm.wx.wy.sigma &lt; Inf</code> .
includeData	Logical indicating whether the original data is included to the model output. Using <code>FALSE</code> can be used to save memory.
calculateZ	Logical indicating whether an expectation of the latent variable $Z$ is included in the model output. Otherwise the expectation can be calculated with <code>getZ</code> or <code>z.expectation</code> . Using <code>FALSE</code> speeds up the calculation of the dependency model.
verbose	Follow procedure by intermediate messages.

## Details

The `fit.dependency.model` function fits the dependency model  $X = N(W\$X * Z, \text{phi}\$X)$ ;  $Y = N(W\$Y * Z, \text{phi}\$Y)$  with the possibility to tune the model structure and parameter priors.

In particular, the dataset-specific covariance structure  $\text{phi}$  can be defined; non-negative priors for  $W$  are possible; the relation between  $W\$X$  and  $W\$Y$  can be tuned. For a comprehensive set of examples, see the example scripts in the `tests/` directory of this package.

Special cases of the model, obtained with particular prior assumptions, include probabilistic canonical correlation analysis (`pcca`; *Bach & Jordan 2005*), probabilistic principal component analysis (`ppca`; *Tipping & Bishop 1999*), probabilistic factor analysis (`pfa`; *Rubin & Thayer 1982*), and a regularized version of canonical correlation analysis (`pSimCCA`; *Lahti et al. 2009*).

The standard probabilistic PCA and factor analysis are methods for a single data set ( $X \sim N(WZ, \text{phi})$ ), with isotropic and diagonal covariance ( $\text{phi}$ ) for `pPCA` and `pFA`, respectively. Analogous models for two data sets are obtained by concatenating the two data sets, and performing `pPCA` or `pFA`.

Such special cases are obtained with the following choices in the `fit.dependency.model` function:

**pPCA** `marginalCovariances = "identical isotropic"` (*Tipping & Bishop 1999*)

**pFA** `marginalCovariances = "diagonal"` (*Rubin & Thayer 1982*)

**pCCA** `marginalCovariances = "full"` (*Bach & Jordan 2005*)

**pSimCCA** `marginalCovariances = "full"`, `priors = list(Nm.wxwy.mean = I, Nm.wxwy.sigma = 0)`.

This is the default method, corresponds to the case with  $W\$X = W\$Y$ . (*Lahti et al. 2009*)

**pSimCCA with T prior** marginalCovariances = "isotropic", priors = list(Nm.wxwy.mean = 1, Nm.wx.wy.sigma = 1)  
(Lahti et al. 2009)

To avoid computational singularities, the covariance matrix phi is regularised by adding a small constant to the diagonal.

## Value

[DependencyModel](#)

## Author(s)

Olli-Pekka Huovilainen <ohuovila@gmail.com> and Leo Lahti <leo.lahti@iki.fi>

## References

- Dependency Detection with Similarity Constraints, Lahti et al., 2009 Proc. MLSP'09 IEEE International Workshop on Machine Learning for Signal Processing, <http://arxiv.org/abs/1101.5919>
- A Probabilistic Interpretation of Canonical Correlation Analysis, Bach Francis R. and Jordan Michael I. 2005 Technical Report 688. Department of Statistics, University of California, Berkley. <http://www.di.ens.fr/~fbach/probacca.pdf>
- Probabilistic Principal Component Analysis, Tipping Michael E. and Bishop Christopher M. 1999. *Journal of the Royal Statistical Society, Series B*, **61**, Part 3, pp. 611–622. <http://research.microsoft.com/en-us/um/people/cmbishop/downloads/Bishop-PPCA-JRSS.pdf>
- EM Algorithms for ML Factorial Analysis, Rubin D. and Thayer D. 1982. *Psychometrika*, vol. **47**, no. 1.

## See Also

Output class for this function: [DependencyModel](#). Special cases: ppca, pfa, pcca

## Examples

```
data(modelData) # Load example data X, Y

# probabilistic CCA
model <- pcca(X, Y)

# dependency model with priors (W>=0; Wx = Wy; full marginal covariances)
model <- fit.dependency.model(X, Y, zDimension = 1,
                             priors = list(W = 1e-3, Nm.wx.wy.sigma = 0),
                             marginalCovariances = "full")

# Getting the latent variable Z when it has been calculated with the model
#getZ(model)
```

---

generate.toydata      *Toy data generation*

---

### Description

Generate simulated data which follows the distributional assumptions of the model.

### Usage

```
generate.toydata(N = 100, zDim = 2, xDim = 3, yDim = 3,
  marginal.covariances = "full", priors = NULL)
```

### Arguments

N	Sample size.
zDim	Dimensionality of the latent variable.
xDim	Dimensionality of X data set.
yDim	Dimensionality of Y data set.
marginal.covariances	"full": full covariance matrices for marginal noise (assumed by pCCA); "diagonal": diagonal covariances for marginal noise (pFA); "isotropic": isotropic covariances (pPCA).
priors	Set priors for toydata generation. Use as in <a href="#">fit.dependency.model</a> .

### Details

Assuming normally distributed latent variables for shared component Z, and data-specific components Z<sub>x</sub>, Z<sub>y</sub>. These follow standard multivariate normal distribution N(0, I). The observations X and Y are obtained as  $X = W_x * Z + B_x * Z_x$ ,  $Y = W_y * Z + B_y * Z_y$ .

### Value

List with the following components:

Z, Z <sub>x</sub> , Z <sub>y</sub>	Shared and data-set specific latent variables.
W <sub>x</sub> , W <sub>y</sub> , B <sub>x</sub> , B <sub>y</sub>	Transformation matrices.
X, Y	Data sets.

### Author(s)

Leo Lahti <leo.lahti@iki.fi>

### References

See citation("dmt") for references.

## Examples

```
toy <- generate.toydata(N = 100,  
  zDim = 1, xDim = 3, yDim = 3,  
  marginal.covariances = "full")
```

---

modelData

*An example data set*

---

## Description

Preprocessed gene expression and gene copy number levels of 51 patients in chromosome 17 for 10 genes.

## Usage

```
data(modelData)
```

## Format

**X** Gene expression levels in matrix form. Genes are in columns and samples in rows.

**Y** Gene copy number levels in matrix form. Genes are in columns and samples in rows.

## Source

Integrated gene copy number and expression microarray analysis of gastric cancer highlights potential target genes. Myllykangas et al., *International Journal of Cancer*, vol. **123**, no. **4**, pp. 817–25, 2008.

## Examples

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

---

pfa.neg.log.likelihood

*Likelihood for the models.*

---

## Description

Likelihood functions for the implemented models.

## Usage

```
pfa.neg.log.likelihood(Wvec, phi, X)
```

## Arguments

Wvec	Parameter vector which is converted in the latent covariance structure used in PFA, correspond to $W^*t(W)$ in the model $X = Wz + \text{epsilon}$ .
phi	Marginal covariance in the model $X = Wz + \text{epilon}$ with $\text{epsilon} \sim N(0, \text{phi})$ .
X	Data: features x samples matrix.

## Details

Other likelihood functions will be added later.

## Value

Log-likelihood of the data, given the model parameters.

## Author(s)

Leo Lahti <leo.lahti@iki.fi>

## References

See citation("dmt").

## See Also

fit.dependency.model, pfa

## Examples

```
library(dmt)
# Generate toydata
N <- 100
xdim <- 10
zdim <- 3
toy <- generate.toydata(N = N, zDim = zdim, xDim = xdim, yDim = xdim,
  marginal.covariances = "diagonal")
```

```

# Estimate model parameters
res <- pfa(toy$X, zDimension = zdim)
W <- res@W$total
phi <- res@phi$total
# wtw <- crossprod(t(W)) # is the same as W * t(W)
# Calculate negative log-likelihood for the model
L <- pfa.neg.log.likelihood(W, phi,toy$X)

```

---

plotVar

*Data-specific and shared variance for several dimensionalities*


---

### Description

A function for calculating the captured variations for several different number of retained dimensions. This is a wrapper over `specificVar` and `sharedVar` to help computing both for a range of dimensionalities.

### Usage

```
plotVar(datasets, regcca, dimVector, plot=FALSE)
```

### Arguments

<code>datasets</code>	A list containing the data matrices to be combined. Each matrix needs to have the same number of rows (samples), but the number of columns (features) can differ. Each row needs to correspond to the same sample in every matrix.
<code>regcca</code>	Output of <code>regCCA</code> function, containing the solution of the generalized CCA.
<code>dimVector</code>	A list of dimensions for which the retained variations are to be computed.
<code>plot</code>	A logical variable with default value <code>FALSE</code> . If the value is <code>TRUE</code> , the functions creates a plot of the output.

### Details

The function uses `specificVar` and `sharedVar` to do all the computation. The purpose of this function is to provide an easy way to visualize the properties of the reduced-dimensional representation created by `drCCA`. The function also estimates the same quantities for PCA of concatenated feature vectors to illustrate the difference to optimal linear model based on preserving the total variation in the whole collection of data sets.

### Value

The function returns the data-specific and shared variance for the given values of dimensions in a list. The list has four components.

<code>pw_cca</code>	A vector with values as shared variances captured by <code>drCCA</code> for the given dimensions
---------------------	--

pw_pca	A vector with values as shared variances captured by PCA for the given dimensions
within_cca	A vector with values as data-specific variances captured by drCCA for the given dimensions
within_pca	A vector with values as data-specific variances captured by PCA for the given dimensions

**Author(s)**

Abhishek Tripathi, Arto Klami

**References**

Tripathi A., Klami A., Kaski S. (2007), Simple integrative preprocessing preserves what is shared in data sources, *submitted for publication*.

**See Also**

[sharedVar](#), [specificVar](#)

**Examples**

```
# data(expdata1)
# data(expdata2)
# r <- regCCA(list(expdata1,expdata2))#

# plotVar(list(expdata1,expdata2),r,c(1:2),4)
```

---

regCCA

*Generalized Canonical Correlation Analysis*

---

**Description**

Solve generalized CCA. Contains a possibility to regularize the solution to reduce the effect of noise.

**Usage**

```
regCCA(datasets, reg=0)
```

### Arguments

datasets	A list containing the data matrices to be analyzed. Each matrix needs to have the same number of rows (samples), but the number of columns (features) can differ. Each row needs to correspond to the same sample in every matrix.
reg	Regularization parameter for the whitening step used to remove data-set specific variation. The value of parameter must be between 0 and 1. The default value is set to 0, which means no regularization will be used. If a non-zero value is given it means that some of the dimensions with the lowest variance are ignored when whitening. In more terms, the dimensions whose total contribution to sum of eigenvalues of the covariance matrix of each data set is below reg will not be used for the whitening.

### Details

The function implements generalized CCA by explicitly whitening the data sets and then performing a principal component analysis on the collection of whitened data sets, instead of directly solving the generalized eigenproblem. Singular value decomposition is used for both the whitening and the PCA phase, and row-wise mean values of each data set are removed before whitening.

### Value

The function returns a list with following components

eigval	Generalized canonical correlations. In case of two data sets (eigval-1) would give the correlations.
eigvecs	List of projection matrices, one for each data set. Each projection matrix is a N times m matrix where N is the number of samples and m is the total number of dimensions in all of the data sets.
proj	Projection of the original data sets by the corresponding projection matrices.
meanvec	An array containing columnwise mean vectors for each data matrix
white	An array of whitening matrices for each data set. This might not be of user interest but this value is used as input in other functions in the package.

The function also prints whether regularization was used or not.

### Author(s)

Abhishek Tripathi <abhishektripathi.at@gmail.com>, Arto Klami

### References

- Hotelling H. (1936), Relations between two sets of variables, *Biometrika*, **28**, 321-327.
- Kettenring J.R. (1971), Canonical Analysis of several sets of variables, *Biometrika*, **58:3**, 433-451.
- Tripathi A., Klami A., Kaski S. (2007), Simple integrative preprocessing preserves what is shared in data sources.



**See Also**

cancor,prcomp,svd

**Examples**

```
# data(expdata1)
# data(expdata2)

#performing regCCA
# test <- regCCA(list(expdata1,expdata2),0) #list of result is stored in test

# test$eigval #generalized canonical correlations
# test$eigvecs #gCCA components
# test$proj #projection of data onto gCCA components
# test$meanvec #array of columnwise mean vectors for each matrix
# test$white # array of whitening matrix
```

---

sharedVar

*Shared variation retained in the combined drCCA representation*

---

**Description**

A function for estimating the amount of shared variation (i.e. variation that is common to more than one data set) retained in the combined data set of given dimensionality.

**Usage**

```
sharedVar(datasets,regcca,dimension,pca=FALSE)
```

**Arguments**

datasets	A list containing the data matrices to be combined. Each matrix needs to have the same number of rows (samples), but the number of columns (features) can differ. Each row needs to correspond to the same sample in every matrix.
regcca	Output of regCCA function, containing the solution of the generalized CCA.
dimension	The number of dimensions of projected data to be used
pca	A logical variable with default value FALSE. If the value is TRUE, the pairwise variation will also be calculated for the PCA projected data, where PCA is performed on the columnwise concatenation of the given data sets.

## Details

The function estimates the amount of shared information retained in a previously calculated drCCA solution. It calculates the shared variation between all pairs of the data sets returned from drCCA combined data for a particular dimensionality. The function also calculates the same quantities for the original data and for the simple PCA projection of the concatenation of data sets. This can be used as a comparison value. If the full dimensionality of drCCA projection or the PCA projection is used, the sum of all pairs of shared variations will be the same. The mean of shared variations for drCCA and PCA is estimated, normalized in a way that the value for original data sets will be 1. A good result will have value greater than 1. For details please refer to the reference below.

## Value

A list of following elements is returned

oo	A matrix containing the pairwise shared variations for original data sets
cc	A matrix containing the pairwise shared variations for a drCCA projection of given dimensions
pc	A matrix containing the pairwise shared variations for a PCA projection of given dimensions, if <code>pca = TRUE</code> is given
mcca	Mean of shared variation between all pairs for drCCA
mpca	Mean of shared variation between all pairs for PCA, if <code>pca = TRUE</code> is given

## Author(s)

Abhishek Tripathi <abhishektripathi.at@gmail.com>, Arto Klami

## References

Tripathi A., Klami A., Kaski S. (2007), Simple integrative preprocessing preserves what is shared in data sources.

## See Also

[specificVar](#)

## Examples

```
# data(expdata1)
# data(expdata2)
# r <- regCCA(list(expdata1,expdata2))

# sharedVar(list(expdata1,expdata2),r,4)
```

---

specificVar	<i>Data-specific variation retained in the combined drCCA representation</i>
-------------	--

---

### Description

A function for estimating the amount of data-set specific variation (i.e. variation that is not present in any of the other data sets) retained in the combined data set of given dimensionality.

### Usage

```
specificVar(datasets, regcca, dim, pca=FALSE)
```

### Arguments

datasets	A list containing the data matrices to be combined. Each matrix needs to have the same number of rows (samples), but the number of columns (features) can differ. Each row needs to correspond to the same sample in every matrix.
regcca	Output of regCCA function, containing the solution of the generalized CCA.
dim	The number of dimensions of projected data to be used
pca	A logical variable with default value FALSE. If the value is TRUE, the data-specific variation will also be calculated for the PCA projected data, where PCA is performed on the columnwise concatenation of the given data sets.

### Details

The function estimates the amount of data-specific information retained in a previously calculated drCCA solution. The function uses SVD to estimate the variance of each data set in the drCCA projection of the given dimensions. Data-specific variance is defined as the sum of singular values for the covariance matrix of a data set. The value is normalized so that the variation for each of the original data sets is 1. The average of the data-specific variances in the projection is also calculated. A solution truly focusing on the dependencies usually has a value that grows roughly linearly when the number of dimensions is increased. The function can also be used to estimate the same quantity for simple PCA projection of the concatenation of the data sets. This can be used as a comparison value. For details, please check the reference.

### Value

The function returns a list of following values

cc	Data Specific variation for a drCCA projection of given number of dimensions
pc	A vector containing the data-specific variations for a PCA projection of given dimensions, if <code>pca = TRUE</code> is given
mcca	Mean of data-specific variations for a drCCA projection
mpca	Mean of data-specific variation for a PCA projection, if <code>pca = TRUE</code> is given

**Author(s)**

Abhishek Tripathi <abhishektripathi.at@gmail.com>, Arto Klami

**References**

Tripathi A., Klami A., Kaski S. (2007), Simple integrative preprocessing preserves what is shared in data sources.

**See Also**

[sharedVar](#)

**Examples**

```
# data(expdata1)
# data(expdata2)
# r <- regCCA(list(expdata1,expdata2))

# specificVar(list(expdata1,expdata2),r,4)
```

---

z.expectation

*Expectation of the latent variable.*

---

**Description**

Calculates expectation of the latent variable, given data and model parameters.

**Usage**

```
z.expectation(model, X, Y = NULL)
```

**Arguments**

model	<a href="#">DependencyModel</a>
X	Data set X.
Y	Optional, second data set. Used in dependency models where two co-occurring observations are assumed to stem from a shared latent variable.

**Value**

Returns a matrix (latent features x samples).

**Author(s)**

Leo Lahti <leo.lahti@iki.fi>

## **References**

See `citation("dmt")` for references.

## **Examples**

```
library(dmt)
data(modelData)
res <- pfa(X, Y, zDimension = 2)
```

# Index

- \*Topic **classes**
  - DependencyModel-class, 5
- \*Topic **datasets**
  - expdata, 7
  - modelData, 12
- \*Topic **maths**
  - centerData, 2
  - dependency.score, 4
- \*Topic **math**
  - fit.dependency.model, 8
- \*Topic **multivariate**
  - drCCAcombine, 6
  - plotVar, 14
  - regCCA, 15
  - sharedVar, 17
  - specificVar, 19
- \*Topic **utilities**
  - centerData, 2
  - compare.estimate.and.truth, 3
  - dependency.score, 4
  - generate.toydata, 11
  - pfa.neg.log.likelihood, 13
  - z.expectation, 20
- centerData, 2
- compare.estimate.and.truth, 3
- dependency.score, 4
- DependencyModel, 10, 20
- DependencyModel-class, 5
- drCCAcombine, 6
- expdata, 7
- expdata1 (expdata), 7
- expdata2 (expdata), 7
- fit.dependency.model, 5, 8, 11
- generate.toydata, 3, 11
- getModelMethod (DependencyModel-class), 5
- getModelMethod, DependencyModel-method (DependencyModel-class), 5
- getParams (DependencyModel-class), 5
- getParams, DependencyModel-method (DependencyModel-class), 5
- getPhi (DependencyModel-class), 5
- getPhi, DependencyModel-method (DependencyModel-class), 5
- getScore (DependencyModel-class), 5
- getScore, DependencyModel-method (DependencyModel-class), 5
- getW (DependencyModel-class), 5
- getW, DependencyModel-method (DependencyModel-class), 5
- getWindowSize (DependencyModel-class), 5
- getWindowSize, DependencyModel-method (DependencyModel-class), 5
- getZ (DependencyModel-class), 5
- getZ, DependencyModel-method (DependencyModel-class), 5
- modelData, 12
- pcca, 5
- pcca (fit.dependency.model), 8
- pfa, 5
- pfa (fit.dependency.model), 8
- pfa.neg.log.likelihood, 13
- plotVar, 14
- ppca, 5
- ppca (fit.dependency.model), 8
- regCCA, 6, 7, 15
- sharedVar, 15, 17, 20
- specificVar, 15, 18, 19
- X (modelData), 12
- Y (modelData), 12
- z.expectation, 20