Imputing missing values using the pcaMethods package

Wolfram Stacklies and Henning Redestig
CAS-MPG Partner Institute for Computational Biology (PICB)
Shanghai, P.R. China
and
Max Planck Institute for Molecular Plant Physiology
Potsdam, Germany
http://bioinformatics.mpimp-golm.mpg.de/

November 24, 2026

1 Missing value imputation

One application for missing value robust principal component analysis is that it effectively can be used to impute the missing values and thus obtain an estimated complete data set. The pcaMethods package was partly written with this application in mind.

PCA is a way of creating a model of a matrix, X, by defining two parameter matrices, the scores, T, and the loadings, P, which together have less values than the original matrix but when multiplied with each other well reconstruct the original matrix. I.e.:

$$X = 1 \times \bar{x} + TP' + E$$

where E is the error matrix and $1 \times \bar{x}$ denotes the original variable averages. Now if X contains missing values but we still are able to get complete estimates of P and T than we can use:

$$\hat{X} = 1 \times \bar{x} + TP'$$

as an estimate for $x_{i,j}$ if $x_{i,j}$ is missing.

This is can be done as the following example illustrates. First we attach the metabolite data set with missing values.

```
> library(pcaMethods)
```

```
> data(metaboliteData)
> mD <- metaboliteData
> sum(is.na(mD))
```

```
[1] 419
```

Now we get the estimated data set by using PPCA and three principal components.

```
> pc <- pca(mD, nPcs=3, method="ppca")
> imputed <- completeObs(pc)
```

If we compare with the original values we see that the error is rather low.

```
> data(metaboliteDataComplete)
> mdComp <- metaboliteDataComplete
> sum((mdComp[is.na(mD)] - imputed[is.na(mD)])^2) / sum(mdComp[is.na(mD)]^2)
```

```
[1] 0.102481
```

When using a different PCA algorithm, we get different performance.

```
> imputedNipals <- completeObs(pca(mD, nPcs=3, method="nipals"))
> sum((mdComp[is.na(mD)] - imputedNipals[is.na(mD)])^2) / sum(mdComp[is.na(mD)]^2)
```

```
[1] 0.1215919
```

If the data we are interested in was gene expression set of class 'ExpressionSet' we could simply do

```
> library(Biobase)
> data(sample.ExpressionSet)
> exSet <- sample.ExpressionSet
> exSetNa <- exSet
> exprs(exSetNa)[sample(13000, 200)] <- NA
> lost <- is.na(exprs(exSetNa))
> pc <- pca(exSetNa, nPcs=2, method="ppca")
> impExSet <- asExprSet(pc, exSetNa)
> sum((exprs(exSet)[lost] - exprs(impExSet)[lost])^2) / sum(exprs(exSet)[lost]^2)
```

```
[1] 0.03777137
```

Different results will be obtained with different PCA algorithms. Which one to use depends on the general structure of the data set and the imputation performance can be estimated by cross-validation. Please see the 'introduction' vignette on further details on how to use the cross-validation capabilities of this package.