

Package ‘svcR’

September 12, 2009

Version 1.6.2

Date 2009-09-10

Title svcR package

Author Nicolas Turenne <nicolas.turenne@jouy.inra.fr>.

Maintainer Nicolas Turenne <nicolas.turenne@jouy.inra.fr>

Depends R (>= 2.3), spdep, quadprog, ade4, methods

Description svcR implements a support vector machine technique for clustering It includes choice between three kinds of kernel. Original cluster labeling and export of clusters. Format of data require 3 files: matrix, attributes, variables.

LazyLoad Yes

ZipData no

License GPL (>= 2)

Repository CRAN

Date/Publication 2009-09-12 12:18:49

R topics documented:

ClusterEval	2
ClusterEval-class	3
findSvcModel	4
findSvcModel-class	7
kernelMatrix	8
kernelMatrix-class	9
Labelling	10
Labelling-class	12
ModelSV	13
ModelSV-class	14
plot	15

term_att	16
term_mat	17
term_var	18

Index	19
--------------	-----------

ClusterEval	<i>Computation of evaluation</i>
-------------	----------------------------------

Description

Set of functions making evaluation. It requires a class column in data.

Usage

```
## S4 method for signature 'findSvcModel':
ClusterEval.Pvalue(x=NULL)
```

```
## S4 method for signature 'matrix':
DataSampling(DatMat=matrix())
```

```
## S4 method for signature 'findSvcModel':
ClusterEval.crossval(x=NULL)
```

```
## S4 method for signature 'findSvcModel':
ClusterEval.final(x=NULL)
```

```
## S4 method for signature 'findSvcModel':
ClusterEval.scalable(x=NULL)
```

```
## S4 method for signature 'findSvcModel':
ClusterEval.clusterable(x=NULL)
```

Arguments

x	a findSvcModel object
DatMat	iris matrix obtain from a run with findSvcModel function

Details

DataSampling, ClusterEval.crossval, ClusterEval.final, ClusterEval.scalable and ClusterEval.clusterable are based only on iris data. ClusterEval.scalable make evaluation building a matrix adding line by line to current matrix from iris matrix. ClusterEval.clusterable makes evaluation with a set of q paramter values and a set of nu parameter values.

Value

return a numeric or a list of numeric meaning a precision value between 0 and 100 for the method ClusterEval.scalable and ClusterEval.clusterable it return a list of precision values.

Author(s)

Nicolas Turenne - INRA France <nicolas.turenne@jouy.inra.fr>

References

N.Turenne , Some Heuristics to speed-up Support Vector Clustering , technical report 2006, INRA, France <http://migale.jouy.inra.fr/~turenne/svc.pdf>

Examples

```
## exemple with iris data

# generation of the svc model
fmc = findSvcModel.Test();

# catch evaluation value
P = ClusterEval.Pvalue( fmc );
```

ClusterEval-class *Class "ClusterEval"*

Description

reading data, computation of the svc model and cluster labelling

Objects from the Class

Objects can be created by calls of the form `new("ClusterEval", ...)`. or by calling the `ClusterEval` function.

Slots

Precision: Object of class "numeric" contains rate of homogeneity

Author(s)

Nicolas Turenne - INRA France <nicolas.turenne@jouy.inra.fr>

See Also

[ClusterEval](#)

Examples

```
## exemple with iris data

#model computation
fmc = findSvcModel.Test();

#precision assessment
P = ClusterEval.Pvalue(fmc);
```

findSvcModel	<i>Computation of clustering model by support vector machine</i>
--------------	--

Description

SvcR implements a clustering algorithm based on separator search in a feature space between points described in a data space. Data format is defined by an attribute/value table (matrix). The data are transformed within a kernel to a feature space into a unic cluster bounded with a ball radius and support vectors. We can used the radius of this ball in the data space to reconstruct the boundary shaped now in several clusters.

Usage

```
## S4 method for signature 'list':
findSvcModel( x=iris, MetOpt="optimStoch", MetLab="gridLabeling", KernChoice="KernG

## S4 method for signature 'character':
findSvcModel.chargeMatrixHDD( x="term", fileIn="c:/files" )

## S4 method for signature 'list':
findSvcModel.chargeMatrix( x=iris )

## S4 method for signature 'matrix':
findSvcModel.Eval( x=matrix() )

## S4 method for signature 'numeric':
findSvcModel.Test()

## S4 method for signature 'findSvcModel':
getNumPoints( object=new("findSvcModel") )

## S4 method for signature 'findSvcModel':
getClassPoints( object=new("findSvcModel") )

## S4 method for signature 'findSvcModel':
getMatriceK( object=new("findSvcModel") )
```

```
## S4 method for signature 'findSvcModel':
getLagrangeCoeff( object=new("findSvcModel") )
```

Arguments

x	means dataFrame parameter in standard use means dataFrame in chargeMatrix use means DatMat in Eval use, a Matrix given as unic argument
MetOpt	option taking value "optimStoch" (stochastic way of optimization) or "optimQuad" (quadratic way of optimization)
MetLab	option taking value "gridLabeling" (grid labelling) or "mstLabeling" (mst labelling) or "knnLabeling" (knn labelling)
KernChoice	option taking value "KernLinear" (Euclidian) or "KernGaussian" (RBF) or "KernGaussianDist" (Exponential) or "KernDist" (Matrix data as Kernel value)
Nu	kernel parameter
q	kernel parameter
K	number of neighbours on the grid
G	size of the grid
Cx	1st data coordinate to plot for 2D cluster extraction
Cy	2nd data coordinate to plot for 2D cluster extraction
fileIn	path where to find files as "D:\R\library\svcR"
object	a findSvcModel object

Details

The main function of the package is called `findSvcModel`. It takes a data set as first argument. This dataset can be either (1) a `data.frame()` structure, or (2) text files stored on hard disk drive.

In case (1) `data.frame` is classical standard list format (see Iris data).

In case (2) format of 'dataName_mat.txt' (data matrix): 1 1 5.1 1 2 3.5 2 3 1.4 it mean $\text{mat}[1, 1] = 5.1$, $\text{mat}[1, 2] = 3.5$, $\text{mat}[2, 3] = 1.4$

format of 'dataName_att.txt' : X1 X2 it mean X1 is the name of first column of the data matrix, X2 is the name of the second column of the data matrix

format of 'dataName_var.txt' : v1 v2 it mean v1 is the name of first line of the data matrix, v2 is the name of the second line of the data matrix

For the labeling parameter (MetLab) three choice are available: "gridLabeling", "mstLabeling" and "knnLabeling".

For the kernel parameter (KernelChoice) four choices are available: "KernLinear", "KernGaussian", "KernGaussianDist" and "KernDist".

Value

An S4 object of class `findSvcModel` The object is the svc model along with the slots :

`lagrangeCoeff`

lagrange coefficients : `getLagrangeCoeff$A`

Matrice	variables names Matrice\$var, attributes names Matrice\$Att and data Matrice\$Mat
MatriceK	kernel matrix
Data	Data Matrix
MinMaxXY	min max values for first and second coordinates
MisClass	missclassified points
dataFrame	prefix name of data for files decoding
fileIn	data file path
ClassPoints	class of grid points
Cx	x column id of data matrix
Cy	y column id of data matrix
Nu	nu value of the svc model
KNN	knn value for labelling
SizeGrid	size grid for labelling
AroundNullVA	almost null value for lagrange coefficient estimation
NumPoints	value fo grid points

slots can be accessed by `getlagrangeCoeff(object)`, `getMatriceK(object)`, `getClassPoints(object)`
`getNumPoints(object)`

Author(s)

Nicolas Turenne - INRA France <nicolas.turenne@jouy.inra.fr>

References

N.Turenne , Some Heuristics to speed-up Support Vector Clustering , technical report 2006, INRA, France <http://migale.jouy.inra.fr/~turenne/svc.pdf>

Examples

```
## exemple with iris data

MetOpt      = "optimStoch";      # optimisation method with randomization
MetLab      = "gridLabeling";    # grid labelling
KernChoice  = "KernGaussian";    # radial kernel
Nu          = 1.0;
q           = 2000;              # lot of clusters
K           = 1;                 # only 1 nearest neighbour for clustering
Cx = Cy     = 0;                 # we use principal component analysis factors
G           = 20;                # size of the grid for cluster labelling
fileIn      = "";                # fileIn might be such as "D:/R/library/svc/", if NULL it will work on

# usage example with a data frame
data(iris);
fmc = findSvcModel( iris, MetOpt, MetLab, KernChoice, Nu, q, K, G, Cx, Cy);
plot(fmc);
```

```
# usage example with import from textfiles
dataTerm = findSvcModel.chargeMatrixHDD("term", fileIn="");
fmc = findSvcModel( dataTerm, MetOpt , MetLab , KernChoice , Nu, q, K, G, Cx, Cy)
plot(fmc);

#
```

findSvcModel-class *Class "findSvcModel"*

Description

reading data, computation of the svc model and cluster labelling

Objects from the Class

Objects can be created by calls of the form `new("findSvcModel", ...)`. or by calling the `findSvcModel` function.

Slots

lagrangeCoeff: Object of class "list" contains the lagrange coefficients : `lagrangeCoeff$A`

Matrice: Object of class "list" contains variables names `Matrice$var`, attributes names `Matrice$Att` and data `Matrice$Mat`

MatriceK: Object of class "vector" contains kernel matrix

Data: Object of class "matrix" contains Data Matrix

MinMaxXY: Object of class "vector" contains min max values for first and second coordinates

MisClass: Object of class "vector" contains missclassified points

dataFrame: Object of class "list" contains the prefix name of data for files decoding

fileIn: Object of class "character" contains data file path

ClassPoints: Object of class "vector" contains class of grid points

Cx: Object of class "numeric" contains the x column id of data matrix

Cy: Object of class "numeric" contains the y column id of data matrix

Nu: Object of class "numeric" contains nu value of the svc model

KNN: Object of class "numeric" contains knn value for labelling

SizeGrid: Object of class "numeric" contains the size grid for labelling

AroundNullVA: Object of class "numeric" contains almost null value for lagrange coefficient estimation

NumPoints: Object of class "vector" contains value fo grid points

Author(s)

Nicolas Turenne - INRA France <nicolas.turenne@jouy.inra.fr>

See Also[findSvcModel](#)**Examples**

```

## exemple with iris data

MetOpt      = "optimStoch";      # optimisation method with randomization
MetLab      = "gridLabeling";    # grid labelling
KernChoice  = "KernGaussian";    # radial kernel
Nu          = 1.0;
q           = 2000; # lot of clusters
K           = 1;      # only 1 nearest neighbour for clustering
Cx = Cy     = 0;      # we use principal component analysis factors
G           = 20;     # size of the grid for cluster labelling
fileIn     = "";      # fileIn might be such as "D:/R/library/svc/", if NULL it will work on

data(iris)
fmc = findSvcModel( iris, MetOpt, MetLab, KernChoice, Nu, q, K, G, Cx, Cy);

NP = getNumPoints(fmc); # values of the grid

```

kernelMatrix

*Computation of clustering model by support vector machine***Description**

This function implements three kinds of kernel to build a kernel matrix: Euclidian, Radial-Base and Exponential.

Usage

```

## S4 method for signature 'vector':
kernelMatrix.compute(x,
                    SymMat = 1,
                    q      = 1,
                    ncol  = 2,
                    nlin   = 2,
                    KernChoice = 1)

```

Arguments

x	data matrix with a vector format
SymMat	indicator to know if matrix is symmetrical or not
q	q parameter of the model

<code>ncol</code>	number of columns of the data matrix
<code>nlin</code>	number of lines of the data matrix
<code>KernChoice</code>	option taking value "KernLinear" (Euclidian) or "KernGaussian" (RBF) or "Kern-GaussianDist" (Exponential) or "KernDist" (Matrix data as Kernel value)

Value

An S4 object of class `kernelMatrix`. The object is the `svc` model along with the slots :

`matrixKernel` kernel matrix with a vector format
`matrixK` kernel matrix with a matrix format

slots can not be accessed.

Author(s)

Nicolas Turenne - INRA France <nicolas.turenne@jouy.inra.fr>

References

N.Turenne , Some Heuristics to speed-up Support Vector Clustering , technical report 2006, INRA, France <http://migale.jouy.inra.fr/~turenne/svc.pdf>

Examples

```
## exemple with iris data
fmc = findSvcModel.Test();
getMatriceK(fmc); # kernel matrix
```

`kernelMatrix-class` Class "kernelMatrix"

Description

reading data, computation of the `svc` model and cluster labelling

Objects from the Class

Objects can be created by calls of the form `new("kernelMatrix", ...)`. or by calling the `kernelMatrix` function.

Slots

matrixKernel: Object of class "vector" contains kernel matrix with a vector format
matrixK: Object of class "matrix" contains kernel matrix with a matrix format

Author(s)

Nicolas Turenne - INRA France <nicolas.turenne@jouy.inra.fr>

See Also

[kernelMatrix](#)

Examples

```
## exemple with iris data

fmc = findSvcModel.Test();

getMatriceK(fmc); # kernel matrix
```

Labelling

Computation of clustering model by support vector machine

Description

Labelling methods try to identify clusters in a grid.

Usage

```
## S4 method for signature 'findSvcModel':
Labelling.compute(x,
                  MetLab           = 1,
                  MatriceKernel    = MK,
                  MatriceK         = M_K,
                  pp               = vpp,
                  Nu               = 1,
                  G               = 1,
                  q               = 1,
                  ncol            = 1,
                  nlin            = 1,
                  RadiusC         = 2,
                  r               = 2,
                  KernChoice       = 0.01,
                  NbClassInData    = 0.01 )

## S4 method for signature 'findSvcModel':
AdjacencyPP(x, MatriceKernel=matrix(), Vec1=vector(), Vec2=vector())

## S4 method for signature 'findSvcModel':
Adjacency(x, MatriceKernel=matrix())
```

```
## S4 method for signature 'findSvcModel':
MST_labelling(x, MatriceKernel=matrix())

## S4 method for signature 'findSvcModel':
KNN_labelling(x, MatriceKernel=matrix())
```

Arguments

<code>x</code>	a <code>findSvcModel</code> object
<code>MetLab</code>	option taking value "gridLabeling" (grid labelling) or "mstLabeling" (mst labelling) or "knnLabeling" (knn labelling)
<code>MatriceKernel</code>	kernel matrix with vector format
<code>MatriceK</code>	kernel matrix with vector format
<code>pp</code>	full matrix as a vector
<code>Nu</code>	kernel parameter
<code>G</code>	kernel parameter
<code>q</code>	kernel parameter
<code>ncol</code>	number of attributes
<code>nlin</code>	size of variables
<code>RadiusC</code>	model radius
<code>r</code>	residual radius
<code>KernChoice</code>	kernel id
<code>NbClassInData</code>	number of classes in data
<code>Vec1</code>	vector for computing adjacency
<code>Vec2</code>	vector for computing adjacency

Value

An S4 object of class `labelling` The object is the svc model along with the slots :

<code>ClassPoints</code>	class of grid points
<code>NumPoints</code>	value fo grid points

slots can not be accessed.

Author(s)

Nicolas Turenne - INRA France <nicolas.turenne@jouy.inra.fr>

References

N.Turenne , Some Heuristics to speed-up Support Vector Clustering , technical report 2006, INRA, France <http://migale.jouy.inra.fr/~turenne/svc.pdf>

Examples

```
## exemple with iris data  
  
fmc = findSvcModel.Test();  
  
getNumPoints(fmc); # values of the grid
```

Labelling-class *Class "Labelling"*

Description

reading data, computation of the svc model and cluster labelling

Objects from the Class

Objects can be created by calls of the form `new("Labelling", ...)`. or by calling the `Labelling` function.

Slots

ClassPoints: Object of class "vector" contains class of grid points

NumPoints: Object of class "vector" contains value fo grid points

Author(s)

Nicolas Turenne - INRA France <nicolas.turenne@jouy.inra.fr>

See Also

[Labelling](#)

Examples

```
## exemple with iris data  
  
fmc = findSvcModel.Test();  
  
getNumPoints(fmc); # values of the grid
```

Description

Computes kernel matrix, lagrange coefficients, support vectors and radius

Usage

```
## S4 method for signature 'numeric':
ModelSV.compute(x ,
                MatriceKernel = NULL,
                MatriceK      = NULL,
                Nu             = 1,
                nlin           = 1,
                MaxIter        = 2,
                MaxValA        = 2,
                AroundNull     = 0.01,
                AroundNullVA   = 0.01 )

## S4 method for signature 'matrix':
OptimQuadProgWcluster(MatriceKernel ,
                      Nu             = 1,
                      MaxValA        = 2,
                      MinW           = 0.0001 )
```

Arguments

x	method of lagrange computation 1 (stochastic) or 2 (quadratic)
MatriceKernel	kernel matrix with vector format
MatriceK	kernel matrix with vector format
Nu	svc parameter
nlin	number
MaxIter	maximum iteration for coefficients computation
MaxValA	number of neighbours on the grid
AroundNull	almost null parameter
AroundNullVA	almost null parameter for coefficients
MinW	min value for coefficients

Value

An S4 object of class ModelSV The object is the svc model along with the slots :

```
lagrangeCoeff      lagrange coefficients : Vectors YA$A
RadiusC            radius of the hypersphere containing data
SmallR            residu of radius
```

OptimQuadProgWcluster return a list containing lagrange coefficients.
slots can not be accessed.

Author(s)

Nicolas Turenne - INRA France <nicolas.turenne@jouy.inra.fr>

References

N.Turenne , Some Heuristics to speed-up Support Vector Clustering , technical report 2006, INRA, France <http://migale.jouy.inra.fr/~turenne/svc.pdf>

Examples

```
## exemple with iris data

# model computing
fmc = findSvcModel.Test();

L = getlagrangeCoeff(fmc); # catch lagrange coefficients
L$A;                      # lagrange coefficients computed in the model
```

ModelSV-class *Class "ModelSV"*

Description

reading data, computation of the svc model and cluster labelling

Objects from the Class

Objects can be created by calls of the form `new("ModelSV", ...)` or by calling the ModelSV function.

Slots

lagrangeCoeff: Object of class "list" contains the lagrange coefficients : VectorWA\$A
RadiusC: Object of class "numeric" contains radius of the hypersphere containing data
SmallR: Object of class "numeric" contains residu of radius

Author(s)

Nicolas Turenne - INRA France <nicolas.turenne@jouy.inra.fr>

See Also

[ModelSV](#)

Examples

```
## exemple with iris data

fmc = findSvcModel.Test();

L = getlagrangeCoeff(fmc); # catch lagrange coefficients
L$A;                       # lagrange coefficients computed in the model
```

plot

Display function of an svc object

Description

Plot permit to display a grid and cluster, ExportClusters can get cluster in text file. Summary gives statistics about clusters.

Usage

```
## S4 method for signature 'findSvcModel':
plot(x)

## S4 method for signature 'findSvcModel':
ExportClusters(fmc=new("findSvcModel"), NameFile="nf")

## S4 method for signature 'findSvcModel':
Summary(fmc=new("findSvcModel"))

## S4 method for signature 'findSvcModel':
GetClustersTerm(fmc=new("findSvcModel"), term="home")

## S4 method for signature 'findSvcModel':
ShowClusters(fmc=new("findSvcModel"))

## S4 method for signature 'findSvcModel':
GetClusterID(fmc=new("findSvcModel"), Id=1)
```

Arguments

x	findSvcModel object
fmc	findSvcModel object
Id	an integer pointing to a cluster
term	a string naming a variable
NameFile	a string naming export files

Author(s)

Nicolas Turenne - INRA France <nicolas.turenne@jouy.inra.fr>

Examples

```
## exemple with iris data

#generate a cluster model
fmc = findSvcModel.Test();

#plot the cluster
plot(fmc);

#export in text file on path defined with parameter catched by findSvcModel function
ExportClusters(fmc, "Exportfilename");

#give information about basics statistics of clusters (size, attributes mean)
Summary(fmc);

#show all clusters
ShowClusters(fmc);

#show clusters where names of variables contain the string X2
GetClustersTerm(fmc, term="121");

#show a clsuter with a specific id, if id is 0, the variables belongs to none cluster
GetClusterID(fmc, Id=1);
```

term_att

data matrix

Description

this data ...

Usage

data(term_att)

Format

a matrix of data

Source

a matrix of data

References

a matrix of data

term_mat

data matrix

Description

this data ...

Usage

`data(term_mat)`

Format

a matrix of data

Source

a matrix of data

References

a matrix of data

term_var

data matrix

Description

this data ...

Usage

```
data(term_var)
```

Format

a matrix of data

Source

a matrix of data

References

a matrix of data

Index

- *Topic **aplot**
 - plot, 15
- *Topic **arith**
 - ClusterEval, 2
 - kernelMatrix, 8
 - Labelling, 10
- *Topic **array**
 - ClusterEval, 2
 - Labelling, 10
- *Topic **classes**
 - ClusterEval-class, 3
 - findSvcModel-class, 7
 - kernelMatrix-class, 9
 - Labelling-class, 12
 - ModelSV-class, 14
- *Topic **classif**
 - ModelSV, 13
- *Topic **cluster**
 - findSvcModel, 4
 - Labelling, 10
- *Topic **datasets**
 - term_att, 16
 - term_mat, 17
 - term_var, 18
- *Topic **list**
 - plot, 15
- *Topic **optimize**
 - ModelSV, 13
- Adjacency (*Labelling*), 10
- Adjacency, findSvcModel-method (*Labelling*), 10
- AdjacencyPP (*Labelling*), 10
- AdjacencyPP, findSvcModel-method (*Labelling*), 10
- ClusterEval, 2, 3
- ClusterEval-class, 3
- ClusterEval-methods (*ClusterEval*), 2
- ClusterEval.clusterable (*ClusterEval*), 2
- ClusterEval.clusterable, findSvcModel-method (*ClusterEval*), 2
- ClusterEval.crossval (*ClusterEval*), 2
- ClusterEval.crossval, findSvcModel-method (*ClusterEval*), 2
- ClusterEval.final (*ClusterEval*), 2
- ClusterEval.final, findSvcModel-method (*ClusterEval*), 2
- ClusterEval.Pvalue (*ClusterEval*), 2
- ClusterEval.Pvalue, findSvcModel-method (*ClusterEval*), 2
- ClusterEval.scalable (*ClusterEval*), 2
- ClusterEval.scalable, findSvcModel-method (*ClusterEval*), 2
- DataSampling (*ClusterEval*), 2
- DataSampling, matrix-method (*ClusterEval*), 2
- ExportClusters (*plot*), 15
- ExportClusters, findSvcModel-method (*plot*), 15
- findSvcModel, 4, 7
- findSvcModel, character-method (*findSvcModel*), 4
- findSvcModel, integer-method (*findSvcModel*), 4
- findSvcModel, list-method (*findSvcModel*), 4
- findSvcModel, matrix-method (*findSvcModel*), 4
- findSvcModel, missing-method (*findSvcModel*), 4

- findSvcModel, numeric-method
(*findSvcModel*), 4
- findSvcModel-class, 7
- findSvcModel-methods
(*findSvcModel*), 4
- findSvcModel.chargeMatrix
(*findSvcModel*), 4
- findSvcModel.chargeMatrix, list-method
(*findSvcModel*), 4
- findSvcModel.chargeMatrixHDD
(*findSvcModel*), 4
- findSvcModel.chargeMatrixHDD, character-method
(*findSvcModel*), 4
- findSvcModel.Eval (*findSvcModel*),
4
- findSvcModel.Eval, matrix-method
(*findSvcModel*), 4
- findSvcModel.IsError
(*findSvcModel*), 4
- findSvcModel.IsError, numeric-method
(*findSvcModel*), 4
- findSvcModel.Test (*findSvcModel*),
4
- findSvcModel.Test, missing-method
(*findSvcModel*), 4

- getClassPoints (*findSvcModel*), 4
- getClassPoints, findSvcModel-method
(*findSvcModel*), 4
- GetClusterID (*plot*), 15
- GetClusterID, findSvcModel-method
(*plot*), 15
- GetClustersTerm (*plot*), 15
- GetClustersTerm, findSvcModel-method
(*plot*), 15
- getlagrangeCoeff (*findSvcModel*), 4
- getlagrangeCoeff, findSvcModel-method
(*findSvcModel*), 4
- getMatriceK (*findSvcModel*), 4
- getMatriceK, findSvcModel-method
(*findSvcModel*), 4
- getNumPoints (*findSvcModel*), 4
- getNumPoints, findSvcModel-method
(*findSvcModel*), 4

- kernelMatrix, 8, 10
- kernelMatrix-class, 9
- kernelMatrix-methods
(*kernelMatrix*), 8

- kernelMatrix.compute
(*kernelMatrix*), 8
- kernelMatrix.compute, vector-method
(*kernelMatrix*), 8
- KNN_labelling (*Labelling*), 10
- KNN_labelling, findSvcModel-method
(*Labelling*), 10

- Labelling, 10, 12
- Labelling-class, 12
- Labelling-methods (*Labelling*), 10
- Labelling.compute
(*Labelling-class*), 12
- Labelling.compute, findSvcModel-method
(*Labelling*), 10

- ModelSV, 13, 15
- ModelSV-class, 14
- ModelSV-methods (*ModelSV*), 13
- ModelSV.compute (*ModelSV*), 13
- ModelSV.compute, numeric-method
(*ModelSV*), 13
- MST_labelling (*Labelling*), 10
- MST_labelling, findSvcModel-method
(*Labelling*), 10

- OptimQuadProgWcluster (*ModelSV*),
13
- OptimQuadProgWcluster, matrix-method
(*ModelSV*), 13

- plot, 15
- plot, findSvcModel-method (*plot*),
15

- ShowClusters (*plot*), 15
- ShowClusters, findSvcModel-method
(*plot*), 15
- Summary (*plot*), 15
- Summary, findSvcModel-method
(*plot*), 15

- term_att, 16
- term_mat, 17
- term_var, 18