Package ‘msr’

February 20, 2015

Version 0.4.1
Date 2012-4-4
Author Samuel Gerber <sgerber@cs.utah.edu>, Kristi Potter <kpotter@sci.utah.edu>, Oliver Ruebel <ruebel1@llnl.gov>
Maintainer Samuel Gerber <sgerber@cs.utah.edu>
Description Package for discrete Morse-Smale complex approximation based on kNN graph. The Morse-Smale complex provides a decomposition of the domain. This package provides methods to compute a hierarchical sequence of Morse-Smale complicies and tools that exploit this domain decomposition for regression and visualization of scalar functions.
Title Morse-Smale approximation, regression and visualization
Depends R (>= 2.10), Matrix, stats, class, glmnet, stats4, e1071, rgl, RColorBrewer, colorspace
License GPL-2
Repository CRAN
Date/Publication 2012-07-11 10:19:46
NeedsCompilation yes

R topics documented:

msr-package ................................................................. 2
camera_estimation ......................................................... 3
diagonal ................................................................. 3
fourpeaks ............................................................... 4
msc.level.ind ............................................................. 5
msc.lm ............................................................... 6
msc.nn ............................................................... 8
msc.slm ........................................................... 10
msc.sublevels .......................................................... 11
plot.msc ........................................................... 12
predict.msc .......................................................... 14
uci_crime_subset ...................................................... 16
Description

Package for discrete Morse-Smale complex approximation based on kNN graph. The Morse-Smale complex provides a decomposition of the domain. This package provides methods to compute a hierarchical sequence of Morse-Smale complicies and tools that exploit this domain decomposition for regression and visualization of scalar functions.

Details

The core functionality rests on the discrete approximation of the Morse-Smale complex from a sample of a function (see msc.nn, msc.nn.svm, msc.nn.kd).

Based on this functionality the regression approach in [2] (see msc.lm and msc.slm) and the exploratory data analysis approach based on the visualization in [3] (see plot.msc) is implemented.

Author(s)

Samuel Gerber, Kristi Potter, Oliver Ruebel

References


David M. Mount and Sunil Arya ANN library http://www.cs.umd.edu/~mount/ANN/

See Also

msc.nn msc.nn.svm msc.nn.kd predict.mscplot.msc msc.lm msc.elnet msc.slm msc.slm.elnet,
Description

Given two images with point correspondences, the goal is to estimate the translation and rotation of two calibrated cameras. This problem can be formulated as a minimization of the total squared algebraic error:

\[ h(R, t) = f(E) = \sum_i (x_i^T E x'_i)^2 \]

with \( x_i = [x_i^T x_i^T] \) and \( x'_i = [x'_i^T x'_i^T] \) being corresponding points on the image plane defined in the respective camera coordinates. The essential matrix \( E = [t] \times R \) is a 3 x 3 rank-2 matrix. In this formulation, the translation between the two cameras is described by the unit vector \( t \), and the relative camera orientation is defined by the orthogonal rotation matrix \( R \). Both \( t \) and \( R \) are expressed in the coordinate frame of \( x \). Due to the formulation of the problem, \( E \) is guaranteed to have only 5 degrees of freedom: 3 to describe the rotation and 2 to determine the translation up to scale. Hence, \( h \) is defined on a 5D manifold embedded in 9D space. For more detailed information on the definition of this problem, see the manuscript by.

Usage

energy

Author(s)

Samuel Gerber

References

Peter Lindstrom and Mark Duchaineau, Factoring Algebraic Error for Relative Pose Estimation, Lawrence Livermore National Laboratory, LLNL-TR-411194, Mar. 2009

Examples

data(camera_estimation)
summary(energy)

diagonal

Description

Diagonal Function, cosine along the diagonal of a d-dimensional hypercube with exponential envelope orthogonal to diagonal.

\[ f(x) = \frac{1}{2} \cos\left( \frac{\sqrt{d} \langle x, v \rangle}{\sqrt{d} \pi} \right) \exp\left( \frac{-\langle x, v \rangle}{d} \right) \]

with \( v = \frac{1}{\sqrt{d}} \) the unit length diagonal vector.
Value
returns N samples form the diagonal function

Author(s)
Samuel Gerber

Examples

data(diagonal)
d <- diagonal()
d <- diagonal(d = 3, p = 4, N=2000)

Description
Fourpeaks is a two-dimensional, additively separable function of four Gaussian peaks
\[ f(x) = \frac{1}{4} \left( e^{-(x_1 - \frac{1}{4})^2/0.3^2} + e^{-(x_2 - \frac{1}{4})^2/0.3^2} + e^{-(x[1] - \frac{3}{4})^2/0.1^2} + e^{-(x_2 - \frac{3}{4})^2/0.1^2} \right). \]
On \([0, 1]^2\) the function has 4 maxima and 9 minima

Value
returns N samples form the fourpeaks function

Author(s)
Samuel Gerber

Examples

data(fourpeaks)
d <- fourpeaks()
d <- fourpeaks(2000)
d <- fourpeaks(N=2000, phi=pi/4)
**Description**

For a given partition id, compute the indices into ms$x belonging to this partition based on a given Morse-Smale complex msLevel.

**Usage**

```
msc.level.ind(msLevel, pid, addExtrema=TRUE)
```

**Arguments**

- **msLevel**: Morse-Smale complex level object.
- **pid**: Partition id number to compute indices for.
- **addExtrema**: Add the extrema indices of this partition (default TRUE)

**Value**

The indicies into ms$x for crystal index.

**Author(s)**

Samuel Gerber

**References**


**See Also**

`msc.nn`
Examples

data(fourpeaks)
d <- fourpeaks()
ms <- msc.nn(y=d[,1], x=d[, 2:3], knn=10, pLevelP = 0.1)
#compute the indices belonging to partition Id 2 at Morse-Smale persistence
ind <- msc.level.ind(ms$level[[1]], 2)

ms <- msc.nn(y=d[,1], x=d[, 2:3], knn=10, nLevels=10)
#compute the indices belonging to partition Id 2 at Morse-Smale persistence level 3
ind <- msc.level.ind(ms$level[[3]], 2)

---

**msc.lm** 
*Morse Smale Complex Linear Regression*

**Description**

Piecewise linear regression on the decomposition of the domain based on the partition induced by the Morse-Smale complex. For msc.elnet an elastic net is fitted instead of a simple linear regression.

For prediction the linear model are either averaged based on weighting the contributions from each partition for a predicting point or predicted based on the linear model corresponding to the highest partition probability. The weights for each partition are computed depending on the underlying Morse-Smale complex type (see msc.nn). The functions can be called with msc.nn without predictive capacities, then prediction of unseen data is not supported.

**Usage**

```r
msc.lm(ms, nfold = 10, modelSelect=FALSE, blend=FALSE, verbose=FALSE)
msc.elnet(ms, nfold = 10, blend=FALSE)
```

**Arguments**

- **ms**: A Morse-Smale complex object, see msc.nn.
- **nfold**: Number of folds for crossvalidation, used for selecting an appropriate persistence level if the underlying Morse-Smale complex objects has multiple levels.
- **modelSelect**: Do a forward stepwise model selection for each linear model (for each partition there is on linear model)
- **blend**: Use blending for model predictions. FALSE results in piecewise linear model.
- **verbose**: Print model fitting information

**Value**

An object of class c("msc.lm") or c("msc.elnet"), that can be used for prediction with predict. The object c("msc.lm") has the following components:

- **ms**: The Morse-Smale complex, see msc.nn
- **lms**: The linear models and crossvalidation results for each level in ms.
blend Use blending for model prediction.

The object c("msc.elnet") has the following components:

ms The Morse-Smale complex, see msc.nn
elnet The elastic net models and crossvalidation results for each level in ms.

Author(s)

Samuel Gerber

References


See Also

msc.nn, predict.msc.lm, glmnet

Examples

# create Morse-Smale complex regression of fourpeaks2d data set
data(fourpeaks)
d <- fourpeaks()
# build Morse-Smale complex
ms <- msc.nn.svm(y=d[,1], x=d[,2:3], pLevel=0.1, knn = 10)
msr <- msc.lm(ms)
# show selected persistence level by cross validation
msr$ms$predictLevel
# print mean squared crossvalidated error
msr$lms[[msr$ms$predictLevel]]$cv
# predict
fp <- predict(msr, d[,2:3])

# fit an elastic model instead
msr <- msc.elnet(ms)
# prediction for elastic model
fp <- predict(msr, d[,2:3])
**Nearest Neighbor Morse Smale Complex**

**Description**

Compute a hierarchy of Morse-Smale complex of the scattered data x using a nearest neighbor based approach at the requested persistence levels. The persistence is a threshold for merging neighboring extrema. If the difference of lower function value of the extrema and the saddle connecting them is below persistence the extrema are merged. The msc.nn.svm and msc.nn.kd construct Morse-Smale complex that allow probabilistic prediction (using `predict`) of the partition assignment of unseen data points, see also `predict.msc`. The nearest neighbor computation uses the ANN library by David M. Mount and Sunil Arya (http://www.cs.umd.edu/~mount/ANN/).

**Usage**

- `msc.nn(y, x, knn = ncol(x), pLevelP = 0.2, pLevel, nLevels, type = 1, smooth = FALSE, eps=0.01)`
- `msc.nn.kd(y, x, knn = ncol(x)+3, pLevelP = 0.2, pLevel, nLevels, bw, type = 1, smooth = FALSE, eps=0.01)`
- `msc.nn.svm(y, x, knn = 3*ncol(x), pLevelP = 0.2, pLevel, nLevels, cost = 1, type = 1, smooth=FALSE, precompute = FALSE, eps=0.01)`
- `msc.graph(y, x, knn, knnd, nLevels, smooth = FALSE)`

**Arguments**

- **y** Function values at observations x. A numeric vector is expected.
- **x** Observations, a numeric matrix is expected.
- **knn** Number of nearest neighbors for graph computation or matrix with nn indices
- **knnd** Squared nearest neighbor distances has to be same size as knn
- **pLevel** Compute the Morse-Smale complex for a single persistence level given by pLevel. Here, extrema with persistence less than pLevel are ignored.
- **pLevelP** Same as pLevel, but instead of an absolute persistence value, the persistence level is expressed as a percentage of \( \max(y) - \min(y) \)
- **nLevels** If specified computes a hierarchical sequence of Morse-Smale complicies for 2 to nLevels+1 extrema. i.e. from the highest persistence level with a single minimum and maximum to a persitence level with nLevels+1 extrema.
- **type** If 1 use classical persistence for merging based on function value difference at saddle points. For other value use \( R^2 \) measure, i.e. merge partitions that results in the most increase in adj. \( R^2 \) value.
- **smooth** If the data is very noise many extrema are introduced. If smooth is set to true the steepest ascent/descent is not computed based on the raw function values y but based on the function value obtained by averaging the function values of the k nearest neighbors. Effectively, a smoothing of the observed function.
The knn computation is based on an approximation. The parameter \( \text{eps} \) specifies how close the approximation should be, i.e., the ratio of distance to approximate nearest neighbor to true nearest neighbor is at most \( 1 + \text{eps} \) (see the ANN webpage for more details http://www.cs.umd.edu/~mount/ANN/).

- **bw**: Bandwidth for kernel density estimation in each partition.
- **precompute**: Indicates for each level the SVM should be computed and stored. This is useful for speedup if repeated predictions at different levels are required.
- **cost**: Cost for svm for partition classification (see also `svm`).

**Value**

An object of class "msc", "msc.kd" or "msc.svn" with the following components:

- **level**: Containing the Morse-Smale complex at each persistence level.
- **persistence**: Sorted persistence levels at which two extrema merge.
- **predictLevel**: For the `plot.msc`, `predict.msc` methods the persistence level of the Morse-Smale hierarchy at which prediction/plotting is done.
- **nLevels**: Number of persistence levels computed, if pLevel or pLevelP is specified this will be 1.

with "msc$level" the following components:

- **mins**: Indices into x of minima for each partition.
- **maxs**: Indices into x of maxima for each partition.
- **partition**: Partition assignment for each observation in x.
- **partitionSize**: Number of points in each partition.

**Author(s)**

Samuel Gerber

**References**


David M. Mount and Sunil Arya ANN library http://www.cs.umd.edu/~mount/ANN/

**See Also**

`predict.msc`, `plot.msc`, `msc.lm`, `msc.elnet`, `msc.slm`, `msc.slm.elnet`
Examples

data(fourpeaks)
d <- fourpeaks()

# build Morse-Smale complex of m
ms <- msc.nn(y=d[,1], x=d[, 2:3], pLevel=0.1, knn = 15)
ms.kd <- msc.nn.kd(y=d[,1], x=d[, 2:3], pLevel=0.1, knn = 15, bw=0.1)
ms.svm <- msc.nn.svm(y=d[,1], x=d[, 2:3], pLevel=0.1, knn = 15)

# predict partition assignments
p1 <- predict(ms.kd, d[, 2:3])
p2 <- predict(ms.svm, d[, 2:3])

msc.slm  
Morse Smale Complex Simultaneous Linear Regression

Description

Fit a simultaneous linear model using the Morse-Smale decomposition of the domain. For each crystal a new variable is introduced, each observation for the variables is weighted by the weight of belonging to that crystal. The weights are computed depending on the underlying Morse-Smale complex type (see msc.nn).

Usage

msc.slm(ms, nfold = 10, modelSelect = FALSE)
msc.slm.elnet(ms, nfold = 10)

Arguments

ms  
A Morse-Smale complex object, see msc.nn.
nfold  
Number of folds for crossvalidation, used for selecting an appropriate persistence level if the underlying Morse-Smale complex objects has multiple levels.
modelSelect  
Do a forward stepwise model selection for each linear model (for each partition there is one linear model)

Value

An object of class c("msc.slm"), that can be used for prediction with predict.
The object has the following components:

ms  
The Morse-Smale complex, see msc.nn.kd
slm  
The linear model based on the weighted observation and variables for each crystals.
msc.sublevels

Author(s)

Samuel Gerber

References


See Also

predict.msc.slm msc.nn, glmnet

Examples

#create Morse-Smale complex regression of fourpeaks2d data set
data(fourpeaks)
d <- fourpeaks()
#build Morse-Smale complex
ms <- msc.nn.svm(y=d[,1], x=d[, 2:3], pLevel=0.1, knn = 10)
#build model using Morse-Smale decomposition ms
msr <- msc.slm(ms)
#print simultaneous linear model cv error
msr$slm[[msr$ms$predictLevel]]$cv
#predict for all data points
fp <- predict(msr, d[, 2:3])

#use elastic net for fitting instead
msr <- msc.slm.elnet(ms)
fp <- predict(msr, d[, 2:3])

msc.sublevels

Extract levels from Morse Smale Complex

Description

Extract a subset of the levels of the current hierarchical levels of the Morse-Smale complex. This is useful to save computational time for example for building regression models for only a single or smaller range of persistence level of the Morse-Smale hierarchy.

Usage

msc.sublevels(ms, startLevel = ms$predictLevel, endLevel = startLevel)
Arguments

- **ms**: Morse-Smale complex object
- **startLevel**: First Level to include in the new hierarchy
- **endLevel**: Last level to include in the new hierarchy

Value

An object of class `msc` with hierarchy levels from `startLevel` to `endLevel` of the input Morse-Smale object.

Author(s)

Samuel Gerber

References


Examples

```r
data(fourpeaks)
d <- fourpeaks()

#build Morse-Smale complex of m
ms <- msc.nn(y=d[, 1], x=d[, 2:3], nLevels = 15, knn = 15)

#extract levels 9 through 14
ms <- msc.sublevels(ms, 9, 14)
```

---

**plot.msc**

*Visualization of the Morse-Smale Summary of High-Dimensional Scalar Functions*
Description

Visualize the Morse-Smale summary description of a high-dimensional scalar function \( y = f(x) \) with parameters \( x \in \mathbb{R}^n \). For each partition of the Morse-Smale complex, an inverse regression curve is computed that summarizes the domain in that partition. This forms a network of regression curves that connect the extremal points of the function. This network is then embedded in 2D for visualization. The function value of the regression curves is encoded by color and by height in the 3rd dimension for each regression curve. Optional tubes around the regression curves indicate the standard deviation along the curve, representing the approximate extent of the partition. An additional window plots the regression curve for each parameter in \( x \), which allows to examine the behaviour of each partition. Users can select, by mouse-click on the corresponding regression curve, which partitions the plots of the underlying parameters will be shown. In addition, a subset of the parameters \( x \) can be selected using `mscPlotDplotList`.

Usage

```r
## S3 method for class 'msc'
plot(x, drawStdDev=FALSE, span=0.5, nsamples=50,
     plot=TRUE, colormap=0, ...)
## S3 method for class 'msc.kd'
plot(x, drawStdDev=FALSE, span=0.5, nsamples=50,
     plot=TRUE, colormap=0, ...)
## S3 method for class 'msc.svm'
plot(x, drawStdDev=FALSE, span=0.5, nsamples=50,
     plot=TRUE, colormap=0, ...)
## S3 method for class 'mscPlot'
plot(x, drawStdDev=FALSE, axesOn=TRUE, ...)
```

Arguments

- `x` The Morse-Smale complex object for `plot.msc` or the `mscPlot` object for `plot.mscPlot`.
- `drawStdDev` Draw the standard deviation tubes around the plots (default `FALSE`).
- `axesOn` Draw the alignment axes (default `TRUE`).
- `nsamples` Number of samples for piecewise linear approximation to regression curve in each partition.
- `span` Span argument of `loess` for computing regression curves.
- `plot` Show visualization (`TRUE`) or just return the plotting object (`FALSE`).
- `colormap` The choice of colormap. 0 = Blue-Green-Red, 1 = Blue-White-Red, 2 = Purple-White-Green
- `...` additional args have no effect.

Value

An object of class `mscPlot` is used to plot the Morse-Smale summary and allows to manipulate the plotting behaviour. The object `mscPlot` has the following components:

- `geom` which describes the geometry of the summary.
- `scene` which describes the components of the visualization scene.
- `plotList` which defines the specific plots to compare. This is a list containing the plot numbers.
Author(s)
Kristin Potter and Samuel Gerber

References

See Also
msc.nn, msc.nn.svm, msc.nn.kd

Examples

```r
data(fourpeaks)
d <- fourpeaks()

# Build Morse-Smale complex of d
ms <- msc.nn(y=d[,1], x=d[, 2:3], pLevel=0.1, knn = 15)

# Plot the ms summary
p <- plot(ms)

# Change the plots to compare
p$plotList <- c(1)

# Re-plot the summary
plot(p)
```

**predict.msc**

Prediction of partition probabilities of Morse-Smale Complex or regression prediction for Morse-Smale regression models

Description

For **msc.kd** and **msc.svm** compute probabilities for each crystal in the Morse-Smale complex for each point in X based on a kernel density estimator and one-against-all **svm**. For **msc.lm, msc.slm** and **msc.slm.elnet** the prediction based on the fitted regression models.
Usage

```r
## S3 method for class 'msc.kd'
predict(object, newdata, addExtrema=TRUE, ...)
## S3 method for class 'msc.svm'
predict(object, newdata, ...)  
## S3 method for class 'msc.lm'
predict(object, newdata, ...)  
## S3 method for class 'msc.slm'
predict(object, newdata, ...)  
## S3 method for class 'msc.slm.elnet'
predict(object, newdata, ...)
```

Arguments

- `object`: Morse-Smale complex object.
- `newdata`: Observations to predict, if missing the sample form the Morse-Smale complex are used.
- `addExtrema`: Add the extrema indices of this partition (default TRUE)
- `...`: Further arguments are ignored in these functions

Value

For Morse-Smale complex objects a (number of points) x (number of partitions) matrix with probabilities $p(C_i|x)$ of belonging to each crystal. For regression model objects the predicted function values.

Author(s)

Samuel Gerber

References


See Also

`msc.nn msc.lm msc.slm`
Examples

```r
data(fourpeaks)
d <- fourpeaks()
#build Morse-Smale complex
ms <- msc.nn.svm(y=d[,1], x=d[, 2:3], nLevels=15, knn = 10)
#predict partition assignments at level 15
ms$predictLevel = 13
p <- predict(ms, d[, 2:3])
```

uci_crime_subset  

UCI commmunity and crimes subset

Description

Subset of the UCI communities and crime data set—[http://archive.ics.uci.edu/ml/datasets/Communities+and+Crime](http://archive.ics.uci.edu/ml/datasets/Communities+and+Crime). The data set contains 100 variables, with some of the original values with many missing values removed.

From the UCI website: Communities within the United States. The data combines socio-economic data from the 1990 US Census, law enforcement data from the 1990 US LEMAS survey, and crime data from the 1995 FBI UCR.

More detail on the individual variables can be found on the website.

Usage

`crimes`

Author(s)

Samuel Gerber

Examples

```r
data(uci_crime_subset)
summary(crimes)
```
Index

*Topic **classif**
  predict.msc, 14
*Topic **cluster, multivariate, topology, nonparametric, models, nonlinear**
  msc.level.ind, 5
  msc.lm, 6
  msc.nn, 8
  msc.slm, 10
  msc.sublevels, 11
  mss-package, 2
*Topic **datasets**
  camera_estimation, 3
  diagonal, 3
  fourpeaks, 4
  uci_crime_subset, 16
*Topic **hplot, cluster, multivariate, topology, nonparametric, models, nonlinear**
  plot.msc, 12
  camera_estimation, 3
  crimes (uci_crime_subset), 16
  diagonal, 3
  diagonalTopology (diagonal), 3
  energy (camera_estimation), 3
  fourpeaks, 4
  glmnet, 7, 11
  loess, 13
msc, 12
msc (msc.nn), 8
msc.elnet, 2, 9
msc.elnet (msc.lm), 6
msc.kd, 14
msc.level.ind, 5
msc.lm, 2, 6, 9, 14, 15
msc.nn, 2, 5–7, 8, 10, 11, 14, 15
msc.nn.kd, 2, 10, 14
msc.nn.svm, 2, 14
msc.slm, 2, 9, 10, 14, 15
msc.sublevels, 11
msc.svm, 14
mss (mss-package), 2
mss-package, 2
plot.msc, 2, 9, 12
plot.mscPlot (plot.msc), 12
predict, 6, 8, 10
predict.msc, 2, 8, 9, 14
predict.msc.lm, 7
predict.msc.slm, 11
svm, 9, 14
uci_crime_subset, 16