The kknn Package

April 11, 2006

**Title**  Weighted k-Nearest Neighbors

**Version**  1.0.3

**Date**  2006-04-10

**Author**  Klaus Schliep & Klaus Hechenbichler

**Description**  Weighted k-Nearest Neighbors Classification and Regression

**Maintainer**  Klaus Schliep <k.p.schliep@massey.ac.nz>

**Depends**  R(>= 2.0.0)

**License**  GPL version 2 or newer

### R topics documented:

- contr.dummy
- glass
- ionosphere
- kknn
- miete
- simulation
- train.kknn

### Index

1. contr.dummy
2. glass
3. ionosphere
4. kknn
5. miete
6. simulation
7. train.kknn

---

**Description**

Returns a matrix of contrasts.

**Usage**

```r
contr.dummy(n, contrasts = TRUE)
contr.ordinal(n, contrasts = TRUE)
contr.metric(n, contrasts = TRUE)
```
Arguments

n A vector containing levels of a factor, or the number of levels.

contrasts A logical value indicating whether contrasts should be computed.

Details

contr.dummy is standard dummy-coding, contr.metric has the same effect like as.numeric (makes sense of course only for ordered variables). contr.ordinal computes contrasts for ordinal variables.

Value

A matrix with n rows and n-1 columns for contr.ordinal, a matrix with n rows and n columns for contr.dummy and a vector of length n for contr.metric.

Author(s)

Klaus P. Schliep (K.P.Schliep@massey.ac.nz)

References


See Also

contrasts, contr.poly and contr.sdif

Examples

contr.metric(5)
contr.ordinal(5)
contr.dummy(5)

glass

Glass Identification Database

Description

A data frame with 214 observations, where the problem is to predict the type of glass in terms of their oxide content (i.e. Na, Fe, K, etc). The study of classification of types of glass was motivated by criminological investigation. At the scene of the crime, the glass left can be used as evidence... if it is correctly identified!

Usage

data(glass)
Format

A data frame with 214 observations on the following 11 variables.

- **Id**  Id number.
- **RI**  Refractive index.
- **Na**  Sodium (unit measurement: weight percent in corresponding oxide, as are attributes 4-10).
- **Mg**  Magnesium.
- **Al**  Aluminum.
- **Si**  Silicon.
- **K**  Potassium.
- **Ca**  Calcium.
- **Ba**  Barium.
- **Fe**  Iron.

- **Type**  Type of glass: (class attribute) 1 building windows float processed 2 building windows non float processed 3 vehicle windows float processed 4 vehicle windows non float processed (none in this database) 5 containers 6 tableware 7 headlamps

Source

- Creator: B. German, Central Research Establishment, Home Office Forensic Science Service, Aldermaston, Reading, Berkshire RG7 4PN
- Donor: Vina Spiehler, Ph.D., DABFT, Diagnostic Products Corporation

The data have been taken from the UCI Machine Learning Database Repository [http://www.ics.uci.edu/~mlearn/MLRepository.html](http://www.ics.uci.edu/~mlearn/MLRepository.html) and were converted to R format by ⟨K.P.Schliep@massey.ac.nz⟩.

Examples

```r
data(glass)
str(glass)
```

Description

This radar data was collected by a system in Goose Bay, Labrador. This system consists of a phased array of 16 high-frequency antennas with a total transmitted power on the order of 6.4 kilowatts. See the paper for more details. The targets were free electrons in the ionosphere. "Good" radar returns are those showing evidence of some type of structure in the ionosphere. "Bad" returns are those that do not; their signals pass through the ionosphere.

Received signals were processed using an autocorrelation function whose arguments are the time of a pulse and the pulse number. There were 17 pulse numbers for the Goose Bay system. Instances in this database are described by 2 attributes per pulse number, corresponding to the complex values returned by the function resulting from the complex electromagnetic signal.
Usage

```
data(ionosphere)
```

Format

A data frame with 351 observations on the following 35 variables. The first 34 continuous covari-
ables are used for the prediction. The 35th attribute is either "g" ("good") or "b" ("bad") according to
the definition summarized above. This is a binary classification task.

Source

Vince Sigillito (vgs@aplce apl.jhu.edu), Space Physics Group, Applied Physics Laboratory, Johns
Hopkins University, Johns Hopkins Road, Laurel, MD 20723

The data have been taken from the UCI Machine Learning Database Repository
http://www.ics.uci.edu/~mlearn/MLRepository.html
and were converted to R format by ⟨K.P.Schliep@massey.ac.nz⟩.

Examples

```
data(ionosphere)
```

---

kknn

**Weighted k-Nearest Neighbor Classifier**

Description

Performs k-nearest neighbor classification of a test set using a training set. For each row of the
test set, the k nearest training set vectors (according to Minkowski distance) are found, and the
classification is done via the maximum of summed kernel densities. In addition even ordinal and
continuous variables can be predicted.

Usage

```
kknn(formula = formula(train), train, test, na.action = na.omit(),
      k = 7, distance = 2, kernel = "triangular", ykernel = NULL,
      contrasts = c('unordered' = "contr.dummy", ordered = "contr.ordinal"))
```

Arguments

- `formula` A formula object.
- `train` Matrix or data frame of training set cases.
- `test` Matrix or data frame of test set cases.
- `na.action` A function which indicates what should happen when the data contain 'NA's.
- `k` Number of neighbors considered.
- `distance` Parameter of Minkowski distance.
- `kernel` Kernel to use. Possible choices are "rectangular" (which is standard unweighted
  knn), "triangular", "epanechnikov" (or beta(2,2)), "biweight" (or beta(3,3)), "tri-
  weight" (or beta(4,4)), "cos", "inv" and "gaussian".
- `ykernel` A vector containing the 'unordered' and 'ordered' contrasts to use.
This nearest neighbor method expands knn in several directions. First it can be used not only for classification, but also for regression and ordinal classification. Second it uses kernel functions to weight the neighbors according to their distances. In fact, not only kernel functions but every monotonic decreasing function \( f(x) \forall x > 0 \) will work fine.

**Value**

\( kknn \) returns a list-object of class \( kknn \) including the components

- `fitted.values`: Vector of predictions.
- `CL`: Matrix of classes of the \( k \) nearest neighbors.
- `W`: Matrix of weights of the \( k \) nearest neighbors.
- `D`: Matrix of distances of the \( k \) nearest neighbors.
- `prob`: Matrix of predicted class probabilities.
- `response`: Type of response variable, one of \( \text{continuous, nominal or ordinal} \).
- `distance`: Parameter of Minkowski distance.
- `call`: The matched call.
- `terms`: The 'terms' object used.

**Author(s)**

Klaus P. Schliep (K.P.Schliep@massey.ac.nz)
Klaus Hechenbichler

**References**


**See Also**

`train.kknn`, `simulation`, `knn` and `knn1`

**Examples**

```r
library(kknn)
data(iris)
m <- dim(iris)[1]
val <- sample(1:m, size = round(m/3), replace = FALSE,
              prob = rep(1/m, m))
iris.learn <- iris[-val,]
iris.valid <- iris[val,]
iris.kknn <- kknn(Species~., iris.learn, iris.valid, distance = 1,
                  kernel = "triangular")
summary(iris.kknn)
fit <- fitted(iris.kknn)
table(iris.valid$Species, fit)
```
Munich Rent Standard Database (1994)

Description
Many German cities compose so-called rent standards to make a decision making instrument available to tenants, landlords, renting advisory boards and experts. The rent standards are used in particular for the determination of the local comparative rent (i.e. net rent as a function of household size, equipment, year of construction, etc.). For the composition of the rent standards, a representative random sample is drawn from all relevant households, and the interesting data are determined by interviewers by means of questionnaires. The dataset contains the data of 1082 households interviewed for the munich rent standard 1994.

Usage

data(miete)

Format
A data frame with 1082 observations on the following 18 variables.

- **nm** Net rent in DM.
- **wfl** Floor space in m².
- **bj** Year of construction.
- **bad0** Bathroom in apartment? 1: no 0: yes
- **zh** Central heating? 1: yes 0: no
- **ww0** Hot water supply? 1: no 0: yes
- **badkach** Tiled bathroom? 1: yes 0: no
- **fenster** Window type? 1: plain windows 0: state-of-the-art windows
- **kueche** Kitchen type? 1: well equipped kitchen 0: plain kitchen
- **mvdauer** Lease duration in years.
simulation

**wflkat** Floor space category (wfl categorized): 1: less than 50 m², 2: between 51 m² and 80 m², 3: at least 81 m²

**nmqm** Net rent per m².

**rooms** Number of rooms in household.


**adr** Address type: 1: bad, 2: average, 3: good

**wohn** Residential type: 1: bad, 2: average, 3: good

**Source**


The data were converted to R format by ⟨K.P.Schliep@massey.ac.nz⟩.

**Examples**

```r
data(miete)
str(miete)
```

---

**simulation**

*Crossvalidation procedure to test prediction accuracy*

**Description**

*simulation* tests prediction accuracy of regression and/or classification techniques via simulation of different test sets.

**Usage**

```r
simulation(formula, data, runs = 10, train = TRUE, k = 11, ...)
```

**Arguments**

- **formula** A formula object.
- **data** Matrix or data frame.
- **runs** Number of crossvalidation runs.
- **train** A logical value. If TRUE the training procedure for selecting optimal values of k and kernel is performed.
- **k** Number or maximal number of neighbors considered, dependent of choice for train.
- **...** Further arguments passed to or from other methods.

**Value**

A matrix, containing the mean and variance of the misclassification error, the absolute and the squared distances.
Author(s)

Klaus P. Schliep (K.P.Schliep@massey.ac.nz)

References


See Also

kknn and train.kknn

Examples

library(kknn)
data(miete)
simulation(nmqm ~ wfl + bjkat + zh, data = miete, runs = 5, kernel = "triangular", k = 15)
simulation(wflkat ~ nm + bjkat + zh, data = miete, runs = 5)
simulation(zh ~ wfl + bjkat + nmqm, data = miete, runs = 5)

train.kknn

Training kknn

Description

Training of kknn method via leave-one-out crossvalidation.

Usage

train.kknn(formula, data, kmax = 11, distance = 2, kernel = NULL, ykernel = NULL, contrasts = c('unordered' = "contr.dummy", ordered = "contr.ordinal"), ...)

Arguments

formula A formula object.
data Matrix or data frame.
kmax Maximum number of k.
distance Parameter of Minkowski distance.
kernel Kernel to use. Possible choices are "rectangular" (which is standard unweighted knn), "triangular", "epanechnikov" (or beta(2,2)), "biweight" (or beta(3,3)), "triweight" (or beta(4,4)), "cos", "inv" and "gaussian".
ykernel
contrasts A vector containing the 'unordered' and 'ordered' contrasts to use.
... Further arguments passed to or from other methods.
Value

train.kknn returns a list-object of class train.kknn including the components

MISCLASS  Matrix of misclassification errors.
MEAN.ABS  Matrix of mean absolute errors.
MEAN.SQU  Matrix of mean squared errors.
fitted.values List of predictions for all combinations of kernel and k.
best.parameters List containing the best parameter value for kernel and k.
response Type of response variable, one of continuous, nominal or ordinal.
distance Parameter of Minkowski distance.
call The matched call.
terms The `terms` object used.

Author(s)

Klaus P. Schliep ⟨K.P.Schliep@massey.ac.nz⟩

References


See Also

kknn and simulation

Examples

library(kknn)

data(miete)
(train.con <- train.kknn(nmqm ~ wfl + bjkat + zh, data = miete,
  kmax = 25, kernel = c("rectangular", "triangular", "epanechnikov",
  "gaussian", "rank")))
plot(train.con)
(train.ord <- train.kknn(wflkat ~ nm + bjkat + zh, miete, kmax = 25,
  kernel = c("rectangular", "triangular", "epanechnikov", "gaussian",
  "rank")))
plot(train.ord)
(train.nom <- train.kknn(zh ~ wfl + bjkat + nmqm, miete, kmax = 25,
  kernel = c("rectangular", "triangular", "epanechnikov", "gaussian",
  "rank")))
plot(train.nom)

data(glass)
glass <- glass[, -1]
(fit.glass1 <- train.kknn(Type ~ ., glass, kmax = 15, kernel =
  c("triangular", "rectangular", "epanechnikov"), distance = 1))
(fit.glass2 <- train.kknn(Type ~ ., glass, kmax = 15, kernel =
  c("triangular", "rectangular", "epanechnikov"), distance = 2))
plot(fit.glass1)
plot(fit.glass2)
Index

*Topic classif
  contr.dummy, 1
  kknn, 4
  simulation, 7
  train.kknn, 8

*Topic datasets
  glass, 2
  ionosphere, 3
  miete, 6

*Topic design
  contr.dummy, 1
  contr.metric(contr.dummy), 1
  contr.ordinal(contr.dummy), 1
  contr.poly, 2
  contr.sdif, 2
  contrasts, 2
  glass, 2
  ionosphere, 3
  kknn, 4, 7, 9
  knn, 5
  knn1, 5
  miete, 6
  plot.train.kknn(train.kknn), 8
  predict.kknn(kknn), 4
  predict.train.kknn(train.kknn), 8
  print.kknn(kknn), 4
  print.train.kknn(train.kknn), 8
  simulation, 5, 7, 9
  summary.kknn(kknn), 4
  summary.train.kknn(train.kknn), 8
  train.kknn, 5, 7, 8