Package ‘svmpath’

July 14, 2020

Title The SVM Path Algorithm

Date 2020-07-13

Version 0.970

Author Trevor Hastie

Description Computes the entire regularization path for the two-class svm classifier with essentially the same cost as a single SVM fit.

Maintainer Trevor Hastie <hastie@stanford.edu>

Depends kernlab

License GPL-2

URL http://www.jmlr.org/papers/volume5/hastie04a/hastie04a.pdf

NeedsCompilation no

Repository CRAN

Date/Publication 2020-07-14 13:10:02 UTC

R topics documented:

balanced.overlap *simple examples for svmpath*

Description

Datasets for illustrating the svmpath function, that can be plotted while its running

Usage

```
data(svmpath)
```

Format

In each case a list with a component `x` (t column matrix) and a component `y` (vector of +1/-1 values) "Balanced" refers to whether the number of +1s is the same as the -1s. "Overlap" indicates whether the classes are linearly separable. `mixture.data` is a balanced dataset with 100 observations in each class. The others are smaller with between 10-12 obs total.

References

The paper http://www-stat.stanford.edu/~hastie/Papers/svmpath.pdf, as well as the talk http://www-stat.stanford.edu/~hastie/TALKS/svmpathtalk.pdf.

Examples

```
data(svmpath)
attach(balanced.overlap)
svmpath(x,y,trace=TRUE,plot=TRUE)
detach(2)
```

plot.svmpath *plot the svm solution at a step along the path*

Description

produces a plot of the svm solution along the path, and optinally indicates support points

Usage

```
## S3 method for class 'svmpath'
plot(x, step, Size = 60, elbow.show = TRUE, support.show = TRUE, ...)
```

Arguments

`x`	the `svmpath` object
`step`	which step to plot; default is the last step. Use `summary` to see how many steps
`Size`	If the solution is non-linear, this is the gridsize for `countour`
`elbow.show`	Should the points on the elbow be indicated
`support.show`	Should the support points be indicated
`...`	additional arguments to plot, allowing one to change, for example, "main", "xlab" etc

Details

A two-dimensional plot is produced of the SVM solution. Makes sense only if X is two-dimensional. If not, the first two dimensions will be used

Value

A list is returned silently, with the ingredients of the plot

Author(s)

Trevor Hastie

References

The paper http://www-stat.stanford.edu/~hastie/Papers/svmpath.pdf, as well as the talk http://www-stat.stanford.edu/~hastie/TALKS/svmpathtalk.pdf.

See Also

coef.svmpath, svmpath, predict.svmpath, print.svmpath,summary.svmpath

Examples

```
data(svmpath)
attach(balanced.overlap)
fit <- svmpath(x,y,trace=TRUE,plot=FALSE)
plot(fit,step=2)
detach(2)
```

`predict.svmpath` *Make predictions from a "svmpath" object*

Description

Provide a value for `lambda`, and produce the fitted lagrange `alpha` values. Provide values for `x`, and get fitted function values or class labels.

Usage

```
## S3 method for class 'svmpath'
predict(object, newx, lambda, type = c("function", "class",
"alpha", "margin"),...)
```

Arguments

`object`	fitted `svmpath` object
`newx`	values of `x` at which prediction are wanted. This is a matrix with observations per row
`lambda`	the value of the regularization parameter. Note that `lambda` is equivalent to 1/C for the usual parametrization of a SVM
`type`	type of prediction, with default `"function"`. For `type="alpha"` or `type="margin"` the `newx` argument is not required
`...`	Generic compatibility

Details

This implementation of the SVM uses a parameterization that is slightly different but equivalent to the usual (Vapnik) SVM. Here $\lambda = 1/C$. The Lagrange multipliers are related via $\alpha_i^* = \alpha_i/\lambda$, where α_i^* is the usual multiplier, and α_i our multiplier. Note that if `alpha=0`, that observation is right of the elbow; `alpha=1`, left of the elbow; `0<alpha<1` on the elbow. The latter two cases are all support points.

Value

In each case, the desired prediction.

Author(s)

Trevor Hastie

References

The paper http://www-stat.stanford.edu/~hastie/Papers/svmpath.pdf, as well as the talk http://www-stat.stanford.edu/~hastie/TALKS/svmpathtalk.pdf.

See Also

coef.svmpath, svmpath

Examples

```
data(svmpath)
attach(balanced.overlap)
fit <- svmpath(x,y,trace=TRUE,plot=TRUE)
predict(fit, lambda=1,type="alpha")
predict(fit, x, lambda=.9)
detach(2)
```

---

print.svmpath *Print a summary of the SVM path*

Description

print a summary of the fitted svmpath object

Usage

```
## S3 method for class 'svmpath'
print(x, digits, maxsteps, ...)
```

Arguments

`x`	object to be printed
`digits`	number of significant digits (default 6)
`maxsteps`	the number of steps to print; default all
`...`	additional arguments to the generic print function

Value

For each step taken by the algorithm, one or more lines are printed. The step is described in terms of the observation number involved, a coded version of what happened, such as "L->E" meaning "from the Left set" to the "Elbow". Initially all the sets are empty. It gives the margin (sum of the xi), the size of the elbow, and the training error.

Author(s)

Trevor Hastie

References

The paper http://www-stat.stanford.edu/~hastie/Papers/svmpath.pdf, as well as the talk http://www-stat.stanford.edu/~hastie/TALKS/svmpathtalk.pdf.

See Also

coef.svmpath, svmpath, predict.svmpath

Examples

```
data(svmpath)
attach(balanced.overlap)
fit <- svmpath(x,y,trace=TRUE,plot=TRUE)
print(fit)
detach(2)
```

---

radial.kernel *compute the kernel matrix for svmpath*

Description

compute the kernel matrix for svmpath

Usage

```
radial.kernel(x, y=x, param.kernel = 1/p,...)
poly.kernel(x, y=x, param.kernel = 1,...)
```

Arguments

`x`	an n x p matrix of features
`y`	an m x p matrix of features (if omitted, it defaults to `x`)
`param.kernel`	the parameter(s) for the kernel. For this radial kernel, the parameter is known in the fields as "gamma". For the polynomial kernel, it is the "degree"
`...`	unused

Details

For the radial kernel, this computes the function $\exp(-\gamma||x-y||^2)$ for each pair of rows x,y from the input matrices. Here g is param.kernel. For the polynomial kernel, it computes $(xy^T+1)^d$, where d is `param.kernel`.

Value

An n x m matrix.

Author(s)

Trevor Hastie

References

The paper http://www-stat.stanford.edu/~hastie/Papers/svmpath.pdf, as well as the talk http://www-stat.stanford.edu/~hastie/TALKS/svmpathtalk.pdf.

See Also

svmpath

Examples

```
data(svmpath)
attach(balanced.overlap)
fit<-svmpath(x,y,kernel=radial.kernel)
detach(2)
```

`summary.svmpath` *produce a summary of an svmpath object*

Description

printing an svmpath object can produce a lot of lines. The summary methods gives a more concise description by picking out a subset of the steps

Usage

```
## S3 method for class 'svmpath'
summary(object, nsteps = 5, digits = 6, ...)
```

Arguments

`object`	the `svmpath` object
`nsteps`	usually omitted, but can be changed to get longer summaries
`digits`	number of significant digits
`...`	additional arguments to the generic summary function

Details

Uses the `pretty` function to extract the approximately the desired number of steps. Always includes the first and last step.

Value

returns a dataframe with the steps, value of lambda, training error, size of elbow, number of support points, and the sum of the overlaps

Author(s)

Trevor Hastie

References

The paper http://www-stat.stanford.edu/~hastie/Papers/svmpath.pdf, as well as the talk http://www-stat.stanford.edu/~hastie/TALKS/svmpathtalk.pdf.

See Also

coef.svmpath, svmpath, predict.svmpath, print.svmpath

Examples

```
data(svmpath)
attach(balanced.overlap)
fit <- svmpath(x,y,trace=TRUE,plot=TRUE)
summary(fit)
detach(2)
```

---

svmpath — *Fit the entire regularization path for a 2-class SVM*

Description

The SVM has a regularization or cost parameter C, which controls the amount by which points overlap their soft margins. Typically either a default large value for C is chosen (allowing minimal overlap), or else a few values are compared using a validation set. This algorithm computes the entire regularization path (i.e. for all possible values of C for which the solution changes), with a cost a small (~3) multiple of the cost of fitting a single model.

Usage

```
svmpath(x, y, K, kernel.function = poly.kernel, param.kernel = 1, trace,
  plot.it, eps = 1e-10, Nmoves = 3 * n, digits = 6, lambda.min = 1e-04,ridge=0, ...)
```

Arguments

`x`	the data matrix (n x p) with n rows (observations) on p variables (columns)
`y`	The "-1,+1" valued response variable.
`K`	a n x n kernel matrix, with default value `K= kernel.function(x,x)`
`kernel.function`	This is a user-defined function. Provided are `poly.kernel` (the default, with parameter set to default to a linear kernel) and `radial.kernel`
`param.kernel`	parameter(s) of the kernels
`trace`	if `TRUE`, a progress report is printed as the algorithm runs; default is `FALSE`
`plot.it`	a flag indicating whether a plot should be produced (default `FALSE`; only usable with `p=2`
`eps`	a small machine number which is used to identify minimal step sizes

`Nmoves`	the maximum number of moves
`digits`	the number of digits in the printout
`lambda.min`	The smallest value of `lambda = 1/C`; default is `lambda=10e-4`, or `C=10000`
`ridge`	Sometimes the algorithm encounters singularities; in this case a small value of ridge, around 1e-12, can help. Default is `ridge=0`
`...`	additional arguments to some of the functions called by svmpath. One such argument that can be passed is `ridge` (default is 1e-10). This is used to produce "stable" solutions to linear equations.

Details

The algorithm used in `svmpath()` is described in detail in "The Entire Regularization Path for the Support Vector Machine" by Hastie, Rosset, Tibshirani and Zhu (2004). It exploits the fact that the "hinge" loss-function is piecewise linear, and the penalty term is quadratic. This means that in the dual space, the lagrange multipliers will be pieceise linear (c.f. `lars`).

Value

a "svmpath" object is returned, for which there are print, summary, coef and predict methods.

Warning

Currently the algorithm can get into machine errors if `epsilon` is too small, or if `lambda.min` is too small. Increasing either from their defaults should make the problems go away, by terminating the algorithm slightly early.

Note

This implementation of the algorithm does not use updating to solve the "elbow" linear equations. This is possible, since the elbow changes by a small number of points at a time. Future version of the software will do this. The author has encountered numerical problems with early attempts at this.

Author(s)

Trevor Hastie

References

The paper `http://www-stat.stanford.edu/~hastie/Papers/svmpath.pdf`, as well as the talk `http://www-stat.stanford.edu/~hastie/TALKS/svmpathtalk.pdf`.

See Also

print, coef, summary, predict, and FilmPath

Examples

```
data(svmpath)
attach(unbalanced.separated)
svmpath(x,y,trace=TRUE,plot=TRUE)
detach(2)
## Not run: svmpath(x,y,kernel=radial.kernel,param.kernel=.8)
```

Index