

LVQTools Output Usage  
Bachelor project: implementing LVQ in R

Sander Kelders

June 14, 2010

## Introduction

This document is part of a bachelor project which implements several LVQ algorithms in the statistical language R. It describes in short the usage of the output package. The LVQ package produces an object of the form `trainoutput`, `traintestoutput` or `ifoldoutput`. This object can be used with the output package to view the results of the LVQ run. To view the results use the function `show` with the LVQ-output as parameter along with a specification which results to view or use `showAll` with the LVQ-output as only parameter to view all results.

## Usage

1. *LVQoutput*: This is the object in which all the output is stored. It should always be provided when viewing results.
2. *costcurve = FALSE*: Determines if the progress of the costfunction should be plotted.
3. *prototypes = FALSE*: Determines if the endconfiguration of the prototypes should be printed to the screen.
4. *prototypeprogress = FALSE*: Determines if the progress of the prototypes should be printed to the screen.
5. *relevances = FALSE*: Determines if the endconfiguration of the relevances should be plotted.
6. *relevanceprogress = FALSE*: Determines if the progress of the relevances should be plotted.
7. *trainerror = FALSE*: Determines if the ratio of errors, using the trainingset as a test, should be viewed. When `trainoutput` or `traintestoutput` are used the ratio will be printed to the screen. When `ifoldoutput` is used the ratio of all folds will be plotted.
8. *testerror = FALSE*: Determines if the ratio of errors, using the testset as a test, should be viewed. When `trainoutput` or `traintestoutput` are used the ratio will be printed to the screen. When `ifoldoutput` is used the ratio of all folds will be plotted.
9. *trainerrorprogress = FALSE*: Determines if the progress of the error-ratio, using the trainingset as a test, should be plotted.

10. *testerrorprogress = FALSE*: Determines if the progress of the error-ratio, using the testset as a test, should be plotted.
11. *costfold = -1*: When using `nfoldoutput` and viewing the costfunction-progress, determines which fold should be viewed. When -1 is specified all folds will be plotted.
12. *protofold = -1*: When using `nfoldoutput` and viewing the prototypes endconfiguration, determines which fold should be viewed. When -1 is specified all folds will be printed to the screen.
13. *protoprogfold = -1*: When using `nfoldoutput` and viewing the prototypes progress, determines which fold should be viewed. When -1 is specified all folds will be printed to the screen.
14. *relfold = -1*: When using `nfoldoutput` and viewing the endconfiguration of the relevances, determines which fold should be viewed. When -1 is specified all folds will be plotted.
15. *relprogfold = -1*: When using `nfoldoutput` and viewing the progress of the relevances, determines which fold should be viewed. When -1 is specified all folds are plotted.
16. *trainerrorprogfold = -1*: When using `nfoldoutput` and viewing the progress of the error-ratio, using the trainingset as a test, determines which fold should be viewed. When -1 is specified all folds are plotted.
17. *testerrorprogfold = -1*: When using `nfoldoutput` and viewing the progress of the error-ratio, using the testset as a test, determines which fold should be viewed. When -1 is specified all folds are plotted.

## 1 Getters

In order to extract the results from the LVQ-output it is possible to use the `attr`-function provided by R. However several `getters` are also provided for easy acces. When using the `getters` the LVQ-output has to be provided along with the fold, if applicable, with -1 selecting all folds. Every `getter` has the two parameters `LVQout` and `fold`.

1. *getPrototypes*: The endconfiguration of the protoypes.
2. *getRelevances*: The endconfiguration of the relevances.

3. *getPrototypeProgress*: The progress of the prototypes over the whole LVQ-run.
4. *getRelevanceProgress*: The progress of the relevances over the whole LVQ-run.
5. *getCostcurve*: The progress of the costfunction over the whole LVQ-run.
6. *getTrainError*: The error-ratio at the end of the LVQ-run, using the trainingset as a test.
7. *getTestError*: The error-ratio at the end of the LVQ-run, using the testset as a test.
8. *getTrainErrorProgress*: The progress of the error-ratio, using the trainingset as a test.
9. *getTestErrorProgress*: The progress of the error-ratio, using the testset as a test.