FeaLect: **Fea**ture se**Lect**ion by computing statistical scores

Habil Zare

November 1, 2010

Contents

Abstract

FeaLect is a feature selection method by statistically scoring the features. Several random subsets are sampled from the input data and for each random subset, various linear models are fitted using lars method. For each feature, a score is calculated based on the tendency of LASSO in including that feature in the models.

1 Introduction

To build a robust classifier, the number of training instances is usually required to be more than the number of features. In many real life applications such as bioinformatics, natural language processing, and computer vision, many features might be provided to the learning algorithm without any prior knowledge on which ones should be used. Therefore, the number of features can drastically exceed the number of training instances. Many regularization methods have been developed to prevent overfitting and improve the generalization error bound of the predictor in this learning situation. Most notably, Lasso is an ℓ_1-regularization technique for linear regression which has attracted much attention in machine learning and statistics. Although efficient algorithms exist for recovering the whole regularization path [3] for the Lasso, finding a subset of highly relevant features which leads to a robust predictor is an important research question.

A well-known justification of ℓ_1-regularization is that it leads to *sparse* solutions, *i.e.* those with many zeros, and thus performs model selection. Recent research [1,2,5,7] have studied model *consistency* of the Lasso (*i.e.*, if we know the true sparsity pattern of the underlying data-generation process, does the Lasso recover this sparsity pattern when the number of training instances increases?) Analysis in [1,2,7] show that for various decaying schemes of the regularization parameter, Lasso selects the *relevant* features with probability one and *irrelevant* features with positive probability as the number of training instances goes to infinity. If several samples are available from the underlying data distribution, irrelevant features can be removed by simply interesting the set of selected features for each sample. The idea in [2] is to provide such datasets by re-sampling with replacement from the given training dataset using the *bootstrap* method [4].

FeaLect [6] proposes an alternative algorithm for feature selection based on the Lasso for building a robust predictor. The hypothesis is that defining a scoring scheme that measures the "quality" of each feature can provide a more robust selection of features. FeaLect approach is to generate several samples from the training data, determine the best relevance-ordering of the features for each sample, and finally combine these relevance-orderings to select highly relevant features.

2 How to use FeaLect?

FeaLect is an R package source that can be downloaded from The Comprehensive R Archive Network (CRAN). In Linux, it can be installed by the following command:

```
R CMD INSTALL FeaLect_x.y.z.tar.gz
```

where x.y.z. determines the version.

The main function of this package is FeaLect() which is loaded by using the command library(FeaLect) in R.

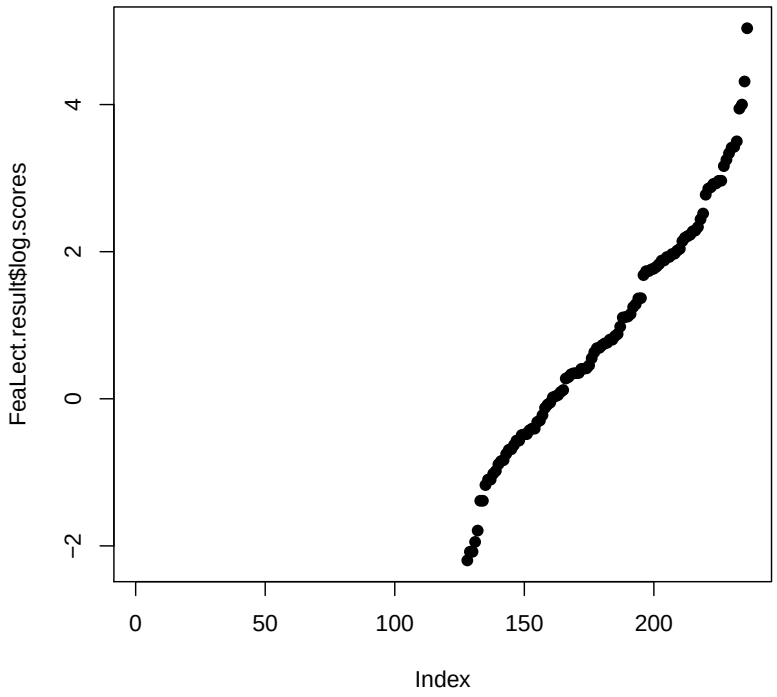
2.1 An example

This example shows how FeaLect can be run to assign scores to features. Here, F is a feature matrix; each column is a feature and each row represents a sample. L is the label vector that contains 1 and 0 for positive and negative samples. We assume L is ordered according to the rows of F.

```
> library(FeaLect)

Design library by Frank E Harrell Jr

Type library(help='Design'), ?Overview, or ?Design.Overview')
to see overall documentation.

> data(mcl_sll)
> F <- as.matrix(mcl_sll[, -1])
> L <- as.numeric(mcl_sll[, 1])
> names(L) <- rownames(F)
> message(dim(F)[1], " samples and ", dim(F)[2], " features.")
> FeaLect.result <- FeaLect(F = F, L = L, maximum.features.num = 10,
+     total.num.of.models = 100, talk = TRUE)

singular information matrix in lrm.fit (rank= 1 ).  Offending variable(s):
linear.scores
singular information matrix in lrm.fit (rank= 1 ).  Offending variable(s):
linear.scores
singular information matrix in lrm.fit (rank= 1 ).  Offending variable(s):
linear.scores
```

The scores are returned in *log.score* element of the output:

```
> plot(FeaLect.result$log.scores, pch = 19)
```



Besides the scores, FeaLect() function computes some other values as well. For instance, the features selected by Bolasso method are also returned as a biproduct without increasing computational cost. Moreover, the package includes some other functions. The input structures and output values are detailed in the package manual.

References

[1] F.˜Bach. Model-consistent sparse estimation through the bootstrap. Technical report, HAL-00354771, 2009.

[2] Francis˜R. Bach. Bolasso: model consistent lasso estimation through the bootstrap. In *ICML '08: Proceedings of the 25th international conference on Machine learning*, 2008.

[3] Bradley Efron, Trevor Hastie, Lain Johnstone, and Robert Tibshirani. Least angle regression. *Annals of Statistics*, 32:407–499, 2004.

[4] Bradley Efron and R.˜J. Tibshirani. *An Introduction to the Bootstrap (Chapman & Hall/CRC Monographs on Statistics & Applied Probability)*. Chapman and Hall/CRC, 1998.

[5] Martin˜J. Wainwright. Sharp thresholds for high-dimensional and noisy sparsity recovery using l1-constrained quadratic programming (lasso). *IEEE Trans. Inf. Theor.*, 55(5), 2009.

[6] Habil Zare, Gholamreza Haffari, Arvind Gupta, and Ryan Brinkman. Statistical analysis of overfitting features. *In preparation.*

[7] Peng Zhao and Bin Yu. On model selection consistency of lasso. *J. Mach. Learn. Res.*, 7:2541–2563, 2006.