A visual toolbox for classification

Amit Meir *, Jonathan Fintzi

Abstract

Classification is one of the most common tasks in statistics and machine learning. Constructing a classifier is highly nontrivial, as the data analyst must optimize predicting performance, often while preserving interpretability, model parsimony, and scientific plausibility. Building a classifier is an iterative process, wherein the data analyst alternates between data exploration, variable selection, and model assessment. The Visual Classification Toolbox was designed to assist an analyst in performing exploratory visualizations to facilitate classifier construction and to visually compare performance of various classifiers. Allowing the analyst to visually inspect the behavior of the classifier with respect to different aspects of the data is central to this task.

Keywords: classification, visualization, Shiny

1 Introduction

Classification is one of the most common tasks in statistics and machine learning. However, classifier construction is highly nontrivial, as the data analyst must optimize predictive performance, often while preserving interpretability, model parsimony, and scientific plausibility. The process of constructing a predictive classifier is an iterative one, wherein the data analyst alternates between data exploration, variable selection, and model assessment. While there exist many effective tools for efficiently exploring patterns within and among variables in a dataset, iteration between variable selection and model assessment is not as well provided for. These steps are often cumbersome since the analyst must perform many tasks each time he wants to explore a different submodel within a class.

In the course of building a classifier, the analyst come to understand not only how the distribution of the response varies according to levels of predictors of interest, but also how the marginal and joint distributions of the predictors in the sample might affect his final model. In their 2004 IEEE paper, Seo and Schneiderman develop an application for visualization of multidimensional data premised on the philosophy that a statistically thorough data exploration first explores each dimension, then explores relationships between the dimensions [Seo and Shneiderman 2004]. The contribution of the application is in allowing the user to explore these univariate and multivariate patterns according to user-defined rankings of features. This philosophy was also adopted by Muhlbacher and Piringer in their tool for building and validating regression models [Muhlbacher and Piringer 2013]. Muhlbacher and Piringer develop a framework for recursively discretizing the marginal and conditional distributions in their sample in order to understand patterns in the response while avoiding distributional assumptions about the model.

We present a tool to enable analysts to use exploratory visualizations to facilitate classifier construction and evaluation. In contrast to the emphasis of Seo and Schneiderman, and of Muhlacher and Piringer, on exploring patterns within the marginal and conditional distributions for classifier construction, our tool emphasizes exploration of distribution of model errors and of the decision boundary in order to facilitate step-wise model construction. We also include basic functionality for evaluating model performance that obviates the need for additional work by the analyst to evaluate classifier performance. Our tool is built using the R Shiny framework, which is a package that enables users to build interactive applications and visualizations on top of their R code. This is an important feature because of the relative ease with which our tool can be extended in the future.

*email:amitmeir@uw.edu

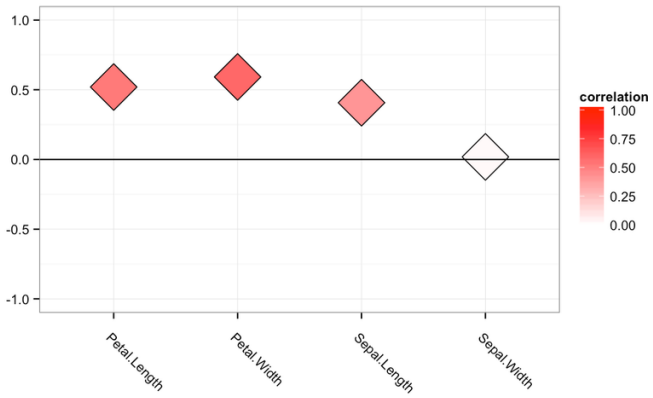

Figure 2: *Main Effect R^2 plot.*

2 Design and Implementation

2.1 Data Input and univariate summaries

The first step in any data analysis project is visual inspection of the data and summary statistics. The first screen the of the Visual Classification Toolbox is the data input screen. Users can upload data sets in several different formats. Once a data set has been uploaded the user may specify the target variable for classification and choose variable with regards to which summary statistics are to be displayed.

Feature Selection

The Visual Classification Toolbox allows the user to include any type of variable in the predictive model, in addition to interaction and quadratic terms. When working with large data sets, the number of possible features that can be included in the model may be overwhelming. Thus, we seek to assist the user in selecting features via two visualizations.

Ideally, we would like to visualize the effect of adding or removing a variable on the wellness of fit. However, especially when dealing with large data sets and complex classifiers this may prove to be computationally expensive. As a proxy, we perform linear regression of each variable on the model residuals and visualize the resulting R^2 from these regressions.

For the main effects, we visualize this measure using a double encoding of color and locations. For the interactions effects, we use a grid where each box is colored according to how much a given interaction or quadratic effect is correlated with the model errors.

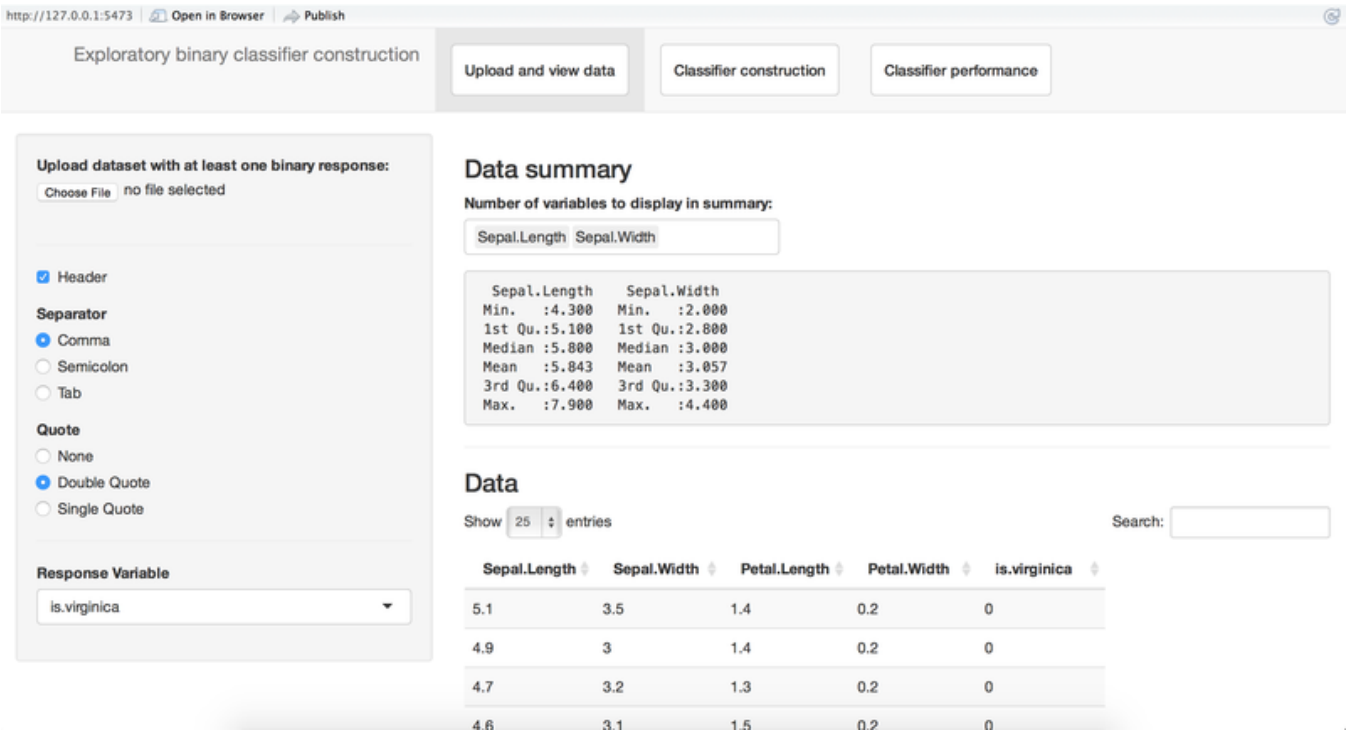

Figure 1: *Data view.*

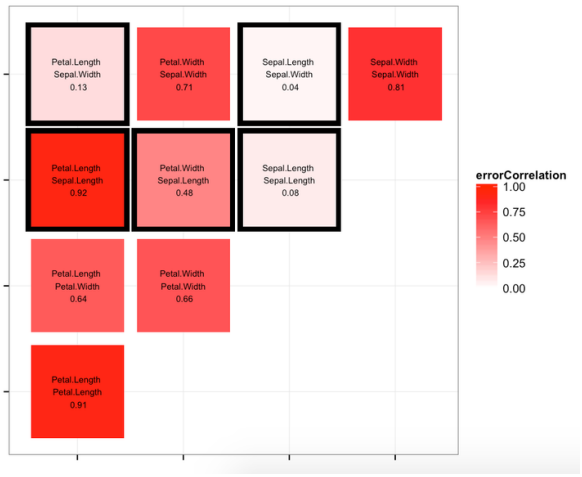

Figure 3: *Interactions R^2 plot.*

Implemented Classification Methods

In the current implementation of the Visual we allow the user to use one of two different classification methods, the LASSO and the Logistic LASSO, as implemented in the glmnet package [Friedman et al. 2010]. We chose to use these two classifiers as they are arguably two of the most fundamental classifiers. However, since our application is based on R, it is very simple to add additional classifiers to the toolbox. Adding a classifier from an existing R package should not take more than a few hours. Once a classifier has been fit, it is possible to adjust the smoothing parameter via a slider interface. Once all variables has been specified, penalty parameter set and a classifier has been chosen, the user can refit the model by pressing on the fit model button.

2.2 Scatter Plots and Classifier Visualization

In order to allow better understanding of the performance of the classifier we provide the user with the ability to draw a scatter plot for any two continuous variables overlaid over the classifier's decision boundary. Since we visualize two dimensional views of the data and the classifier is in general a function of more than two variables, the decision boundary must be projected down to the two dimensional space of the plotted variables. We perform this projection by smoothing the actual decisions made by the algorithm over the plotted subspace using a two dimensional nonparametric regression as implemented by [Hastie 2008]. We enable the user to create small multiples of the decision boundary plot by specifying two variables to facet according to. When faceting is used the projection of the decision boundary is computed for each plot separately.

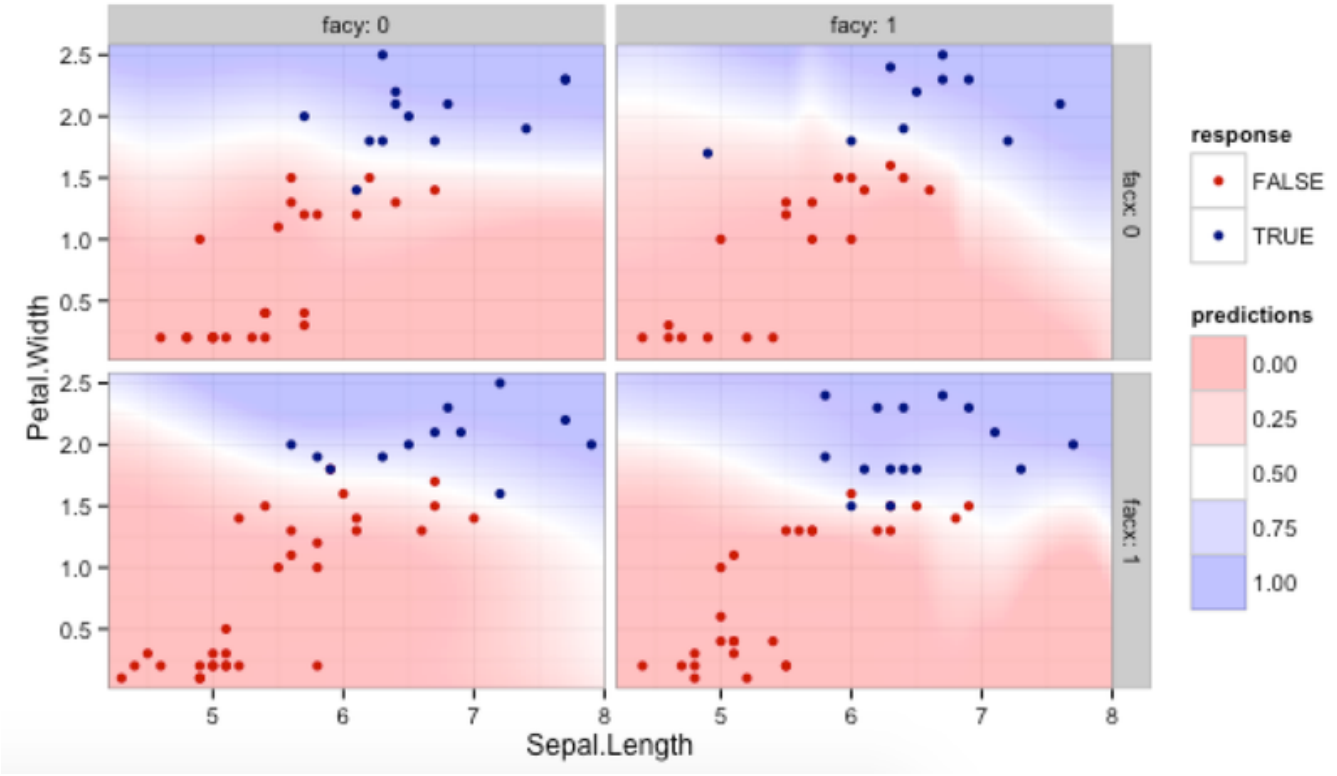

Figure 5: *Main scatter plot visualization with decision regions and faceting.*

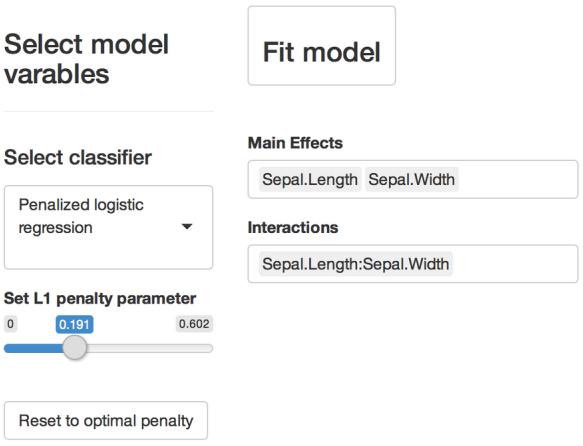

Figure 4: *Model fitting interface.*

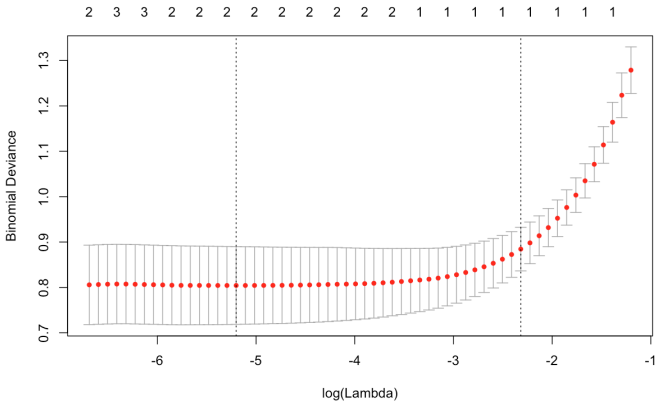

Figure 6: *Cross Validation plot.*

2.3 Predictive Performance Summary

After a model has been fit to the data, the Visual Classification Toolbox assists the user in evaluating the performance of the model by displaying a cross validation plot summary plot and a ROC curve. The cross validation, as produced by [Friedman et al. 2010], gives the error corresponding to different values of penalty, and the number of nonzero coefficients. The ROC plot as produced by [Robin et al. 2011] includes the ROC curve as well as the area under the curve.

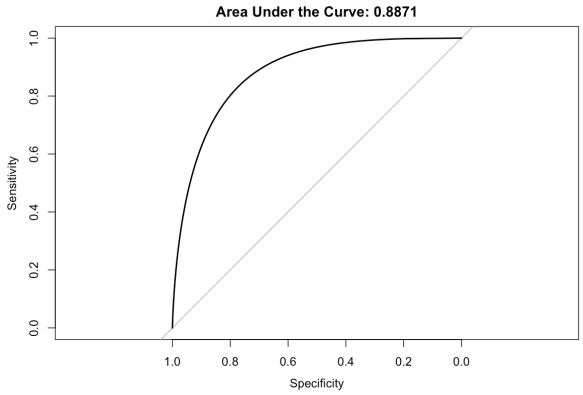

Figure 7: *ROC curve with AUC.*

3 Future Work

We see three avenues for future work: streamlining the model building workflow, extending the tool to allow more integration of user provided R code from within the shiny application, and improving computational performance. One of the shortcomings of the current implementation of our tool is that tracking the variable selection path is entirely left to the user. Adding a visualization to display the selection history of variables in the model along with measures of goodness of fit, such as AIC or BIC, would allow the user to not only track his progress more efficiently, but also to avoid selecting a locally optimum model. Another useful feature that could be added in future work would be functionality allowing the user to export model selection paths and diagnostic plots from a work session.

The major motivation for implementing our tool in Shiny, besides providing us with the opportunity to learn how to use the framework, was the extensibility of Shiny apps through their interaction with R. It is possible to build interactivity into a Shiny app that would allow the user input valid R expressions defining arbitrary functions of the data to use in the classifier, which R would then automatically parse and evaluate. Similarly, users could be allowed to define arbitrary decision rules and visualization parameters that might be relevant to their analyses.

Finally, some of the components of our tool are obvious candidates for computational improvements. In particular, the plots for smoothing the decision boundary across facets could be computed in parallel without tremendous difficulty. Another area where the app could be improved is in caching the results from previously fit models in the current session. This would also enable the user to much more quickly compare new models to previously fit ones.

4 Conclusion

For our final project we implemented a visual tool for classification. This visual tool incorporates several of the tasks faced by a data analyst and assists the analyst in fitting a classifier via a collection visual tools. The Visual Classification Toolbox we as implemented in Shiny, enabling straightforward integration with R and extensibility. We cope with the problem of displaying high dimensional decision regions in two dimension by utilizing a novel approach of smoothing the actual decisions made by the classifier over the plotted subspace.

Despite the current implementation being imperfect, with many of our original ideas not implemented, we hope that this project can serve as a proof of concept or as an inspiration (yes, an inspiration. Because if we can do it, anyone can!) for future, more comprehensive projects.

References

FRIEDMAN, J., HASTIE, T., AND TIBSHIRANI, R. 2010. Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software 33*, 1, 1.

HASTIE, T. 2008. gam: Generalized additive models. *Rpackage version 1*.

MUHLBACHER, T., AND PIRINGER, H. 2013. A partition-based framework for building and validating regression models. *Visualization and Computer Graphics, IEEE Transactions on 19*, 12, 1962–1971.

ROBIN, X., TURCK, N., HAINARD, A., TIBERTI, N., LISACEK, F., SANCHEZ, J.-C., AND MÜLLER, M. 2011. proc: an open-source package for r and s+ to analyze and compare roc curves. *BMC bioinformatics 12*, 1, 77.

SEO, J., AND SHNEIDERMAN, B. 2004. A rank-by-feature framework for unsupervised multidimensional data exploration using low dimensional projections. In *Information Visualization, 2004. INFOVIS 2004. IEEE Symposium on*, IEEE, 65–72.