**Predicting Wine Quality: A Conundrum**

Would you like some cheese with that?

ST 599 Statistical Computing and Big Data-Project 3

June 2, 2014

Matt Edwards, Song Hoa Choi, Gina Shellhammer, Kalbi Zongo

**Introduction**

The goal of our project is to predict the blind taster quality score of a wine based on chemical tests, using the “Wine Quality” data from UCI Machine Learning Repository[[1]](#footnote-1). The response variable is the taster quality score, and there are eleven explanatory variables from various chemical levels in wine. There are two datasets, 4898 white and 1599 red vinho verde wine samples from Northern Portugal, we concentrated on the white wine data.

The taster quality is a discrete scale ranging from 0 to 10, with 0 indicating ‘very bad’ and 10 indicating ‘very excellent’. The median of taster quality in white wine is 6 (n = 2198) with none graded as 0, 1, 2, or 10.

**Description of the machine learning method**

Training and testing sets were created with stratified sampling.

*K*-Nearest Neighbor Regression

*K*-nearest neighbor (KNN) is “a supervised learning algorithm where the result of new instance query is classified based on majority of *k*-nearest neighbor category” (Teknomo, n.s.). The KNN regression uses some measure of distance to find the nearest neighbors in the dataset to the current entry. Entries are ordered by increasing distance, and an “optimal” number, *k*, of nearest neighbors. An inverse distance weighted average is calculated with the *k*-nearest multivariate neighbors. In order to obtain the result, we used the fit function from rminer package in R.

Ordinal Regression

Ordinal regression is one of the general linear models and its formula is similar to logistic binomial regression. The classifications have some inherent order, and for each class, the probability of being in that class or a lower class is calculated, along with the probability of being in a higher class. In other words, for each class, there is a "less than or equal to" probability, and a "greater than" probability. Binary regression models are estimated, and the probability of any one category can be estimated. The prediction is the category with the highest probability. The heavy lifting was done by the ordinal package in R. Model selection using backward and forward was conducted, and both found the same 'best' model.

Multiclass Classification

The classification approach is a multiclass classification algorithm called One vs. All.  The algorithm trains logistic regression parameters for each class--it computes the probability of the class. Similar to Ordinal regression, to predict the class for a new observation, the algorithm picks the class with highest probability, based on the covariates. For classification, three data sets were defined: training set, cross-validation set, and test set. To prevent high bias, we used 4th-degree polynomial regression, resulting in a model with 44 predictors. The downside of this is over-fitting (high variance). To prevent over-fitting, we applied a shrink (penalty) parameter *lambda* to reduce the effect of each predictor. Model selection is conducted to find the best model with using *lambda* values and computing training set error along with cross validation error.

**Summary findings**

K-Nearest Neighbor Regression had a 59.6% success rate. Nothing was allocated to category 3 or 9. Ordinal Regression had a 53.3% success rate. Nothing was allocated to quality categories 3, 8 or 9. Classification had a 54.4% success rate. Again, nothing was allocated to categories 3 or 9. Simply assigning a 5 or 6 to each wine in the test set had a 39% success rate, which we considered our baseline.

**Prediction Results for K Nearest Neighbors and Classification**

**Mac HD:Users:choiso:wine-st599:images:KNNRegression_Results.pdfMac HD:Users:choiso:wine-st599:images:Classification_Results.pdf**

**Discussion including assumptions/limitations**

Assumptions:

Regular logistic regression: individual logistic regression is independent meaning that the probability of all categories does not sum to one. Also, the category with higher probability is more likely to occur than other categories.

Multicolinearity is not an issue with prediction. Multicolinearity blows up standard errors which are very important in estimation, but not so much in prediction.

K-Neighbors: options available for the “search method” for KNN algorithm were not explored. This changes how the hyper-parameters of the algorithm are tuned. Cross validation was not explored for either K-Neighbors or Ordinal regression.

With Ordinal Regression, we assume proportional odds. This means the coefficients, β, stay the same for all categories, and the intercept value αi changes. All explanatory variables have the same weight for all categories.

Limitations:

Regression does not always scale well, and adding covariates can bog down the number of comparisons, especially with model selection. This would apply to all three methods, as they all involved some sort of regression. For scaling, we could try sampling our data, to work with a smaller subset.

For both K-Neighbors and the One-vs-All algorithm, if category distribution is skewed, larger categories can dominates, which is what we see in our results.

One-vs-All does not handle equivalent probabilities well. When two or more categories have the same probability of success, then the approach will just picks the first one. This algorithm can be quite computationally expensive. Our data ran in about 3 minutes, but scalability is an issue for this algorithm.

**References**

Ordinal Regression. Retrieved May 27, 2014 from <http://www.norusis.com/pdf/ASPC_v13.pdf>

Herbrich,R., Graepel, T., & Obermayer, K. (1999). Regression models for ordinal data: A machine learning approach. Retrieved May 29, 2014 from

<http://research.microsoft.com/apps/pubs/default.aspx?id=65632>

Teknomo, K. (n.s.). What is K nearest Neighbors Algorithm? Retrieved May 30, 2014 from

http://people.revoledu.com/kardi/tutorial/KNN/What-is-K-Nearest-Neighbor-Algorithm.html

1. <https://archive.ics.uci.edu/ml/datasets.html> [↑](#footnote-ref-1)