Homework 1 - Applied Machine Learning

Tim Delisle and Sam Raudabaugh

09/15/2015

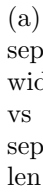

Figure 1: Age and fare distribution for training data

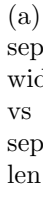

Figure 2: Age and fare distribution for test data

2. Titanic Disaster

For this assignment, we competed in the Titanic dataset challenge on Kaggle. Our goal was to train a logistic regression classifier that uses passenger data, such as age and gender, to predict who survived.

It turns out that the `scikit-learn` package contains a `LogisticRegression` model that we can use for this very purpose. Before we can train such a model, however, we need to clean the data provided by Kaggle, fill in missing values, and select a combination of features for the model to examine.

Preparation of the training and test data occurs within `munge_data()` in the attached code. For the logistic regression training and predicting to work, input data must be numeric, so much of our preparation involves mapping categorical data (i.e. `Sex` and `Port of Embarkation`) to numeric values (`Sex_enum` and `Embarked_enum`).

Next, we fill in missing data - in particular, `Age` and `Fare` - with a dummy value derived from the rest of the dataset. To determine the best dummy values, it is helpful to examine the attached histograms in Figure 1. We see that the distributions of ages and fares are both very skewed, so a median value might be more appropriate than a mean. One sophisticated approach, borrowed from Kaggle's "Getting Started With Python II" guide, is to use the median fare/age for a given gender and passenger's socioeconomic status, which will hopefully better represent typical passengers.

Following Kaggle's "Getting Started With Python II" guide, we con-

structed two new features from the dataset: `FamilySize`, obtained simply by adding together the number of siblings, spouses, parents and children aboard, and `Age*Class`, obtained by multiplying age by the passenger's socioeconomic status (1 for upper class, 2 for middle class, 3 for lower class), since both older and lower class passengers had a lower likelihood of survival.

At each decision step, we test the usefulness of these features with a 10 fold cross-validation technique, which uses the code from Michael Wilber's lecture working with the `cross_validation` module in `scikit-learn` as a starting point. Using `FamilySize` and `Age*Class` in place of `Age` both produced higher cross-validation scores.

We also experimented with pulling out the titles of each passenger's name and categorizing them in different ways. The first strategy was a binary approach with only pedestrian and non-pedestrian titles, while the second strategy used 5 title types: pedestrian, honorary, academic, military, or religious. The latter approach resulted in higher cross-validation scores, but neither approach affected the Kaggle test score. For reference, the attached code includes the second strategy, defined in `get_title_type()`.