Machine Learning 2015: Project 2 - Regression Report

pungast@student.ethz.ch

November 20, 2015

Experimental Protocol

I added nonlinear features to the original features and used a Random Forest classifier to predict the 3 classes.

1 Tools

I used Python with scikit-learn 0.17. Source code is attached to the submission.

2 Algorithm

I used a Random Forest classifier with 40 trees. Parameters are described below.

3 Features

I added the following new features, for each feature x: $logx$, $xlogx$, $\sqrt{x}$, x^3, x^4 and all second-order polynomial combinations of features.

I did not normalise the data because it turned out that neither row-max normalising (dividing all features of each datapoint by the maximum dimension) or column-max normalising (diving all i-th feature values with the maximum i-th feature over all datapoints) improved prediction accuracy in cross-validation.

For visualising the data (for debugging and getting an overview of the data) I used the t-SNE algorithm to map the datapoints into 2 dimensions so that locality information would be conserved. The result is depicted in Figure 1 and shows that the datapoints are quite well linearly separable even in the 2-dimensional space resulting from applying t-SNE.

4 Parameters

I used the following parameters, found in 3-fold cross-validation parameter search (in brackets [] are the parameter values I searched over (all combinations)):

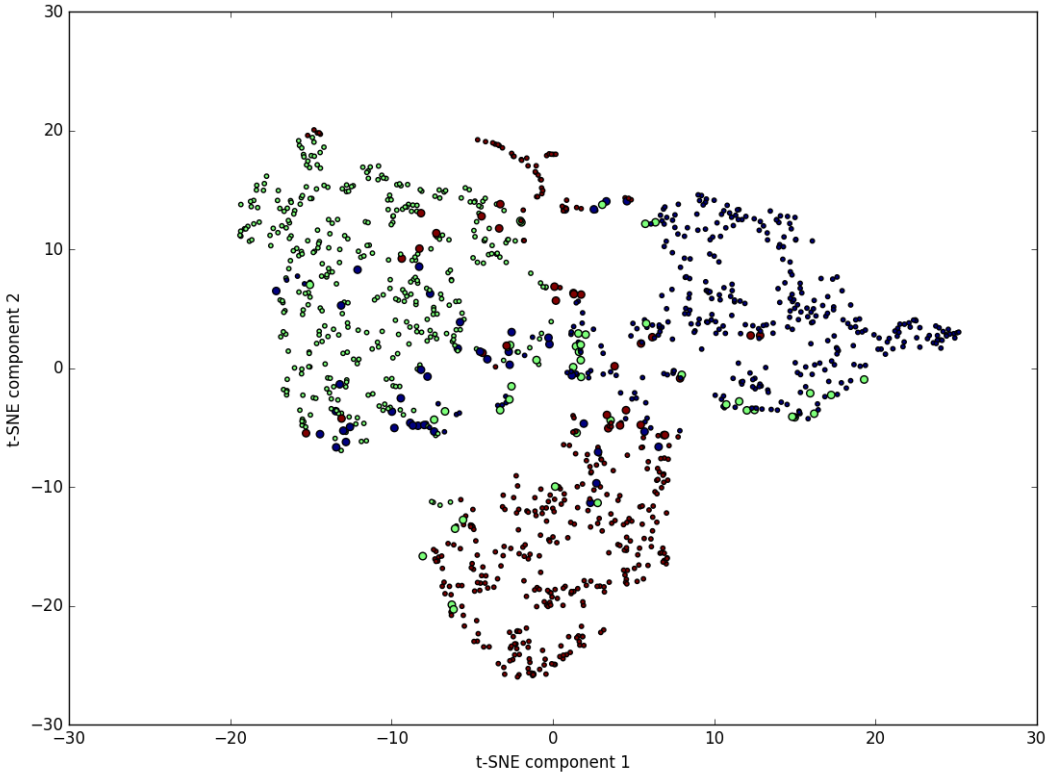

Figure 1: Data after t-SNE transformation into 2 dimensions. Misclassified points are larger than correctly classified points.

- Number of trees in forest: 40 [20, 40, 60, 80, 100, 200]
- Maximum number of features to consider at each split: 2 [1, 2, 3, 4]
- Minimum number of samples required to split an internal node: 5 [1, 3, 5]

5 Lessons Learned

I also tried SVMs with different kernels (RBF, polynomial, linear) but they were all beat by Random Forest in cross-validated parameter searches. It was a surprise to me that even in a quite extensive parameter sweep, SVMs with all kernels I tried were inferior to Random Forests – I would have expected SVMs to be able to learn the nonlinearities better.

A hypothesis: the inferiority of SVMs may be caused by the larger sensitivity of SVMs to the lack of normalisation. However, I tested two kinds of normalising for both SVMs and Random Forests and the prediction performance of SVMs still did not improve.