Notes on Assignment 2 (AI2016)

Julian Togelius

November 30, 2016

Abstract

Some notes on assignment 2 for the AI course.

1 Overview

For assignment 2 you will implement three supervised learning algorithms (k-nearest neighbor, perceptron and ID3 decision tree learning) and two unsupervised learning algorithms (k-means and Agglomerative Nesting).

The task for the supervised learning algorithms is to predict whether a county votes Democratic or Republican. The task for the clustering algorithms is simply to divide US counties into meaningful clusters.

2 The dataset

I have uploaded the full dataset in the Resources section of the class website. The dataset is split into two files:"votes-train.csv", the training set, and "votes-test.csv", the testing set.

The dataset describes voting patterns in the 2016 US presidential election. Each row represents one county. The target class, "democrat", is 1 if the county voted Democratic (Clinton) and 0 if it voted Republican (Trump). (Bear in mind that counties vary drastically in size across the US, and while an overwhelming majority of counties voted Republican, this is not reflective of individual voters.)

The other features/attributes are as follows:

- Population
- Population change (in percent, can be positive or negative)
- Percentage of the population that is 65 years old or older
- Percentage of the population that is black/African-American
- Percentage of the population that is hispanic
- Percentage of the population that has at least a Bachelor's degree
- Mean income (not sure whether this is per person or household)
- Percent of the population living in poverty

- Population density

The dataset is taken from a Kaggle dataset aggregated by Ben Hamner[1]. The original dataset has many more attributes, I removed most of them to make it tractable.

3 Implementing the methods

You can implement the algorithms in any language you want. (Of course, implementations should be done individually.) As always, you have several design choices to make, and it is up to you how you solve these problems to make the code work smoothly.

It is highly advisable that you normalize the data so that all features have values in the range 0 to 1 (or -1 to 1). Some algorithms will be more sensitive to lack of normalization than others.

Given that all the attributes are numerical, you will have to deal with this in building the ID3 decision tree classifier in some way. For example, you could create nominal attributes out of the numerical attributes through binning the values. It is up to you how many bins you choose, and whether to go for equal-width, equal-depth or some other approach.

For the perceptron, remember that if the data is not linearly separable the algorithm will not converge and will "thrash". One way around this is to gradually decrease the learning rate as you train.

4 The report

You should hand in an individual one-page report, together with your source code. The report should discuss the design choices you made in implementing your algorithms, and try to motivate them where possible. It should also present results for all algorithms, so that their performance can be compared. For the supervised learning, you should at least present accuracy. The accuracy should refer to predictions on the test set, after training on the training set. To evaluate the quality of the clustering, you should use a suitable measure such as the Silhouette coefficient.

[1] https://www.kaggle.com/benhamner/2016-us-election