Experiments with k-NN Algorithm

Maria Martinez & Sebastian Cevallos

September 25, 2016

1. Calculate the performance of the perceptron classifier on the 10-fold cross validation of the data (i.e. you should have 10 numbers) with the AveragePerceptronClassifier on the old binary data, i.e. "titanic-train.perc.csv“. Use a reasonable number of iterations based on your experience from last assignment or from a small experiment.
   Also include the average of the 10 folds.
   Because the perceptron algorithm involves randomness (i.e. because it shuffles the examples each round), to do this properly:

   - Generate a 10-fold cross validation. Only do this once for this experiment (i.e. don't keep repeatedly creating new 10-fold cross validations).
   - On each of the splits of the data, run the perceptron 100 times and average those results to get a single value for that split.
   - Repeat this for each of the 10 splits.

   For any of the experiments below for the perceptron classifiers, make sure to follow this procedure to get consistent results.

   Results. The accuracies of each run is reported below, to three significant figures.

Fold number	Accuracy
1	0.749
2	0.733
3	0.519
4	0.860
5	0.830
6	0.760
7	0.802
8	0.844
9	0.724
10	0.741
Total	0.756

2. Calculate the accuracy on the 10 folds on the new non-binary data, i.e. "titanic-train.real.csv". You should notice a pretty big difference here. Why do you think there is such a big difference (you don't have to write your answer)?

Results. The accuracies of each run on the new data is reported below, to three significant figures.

Fold number	Accuracy
1	0.408
2	0.591
3	0.732
4	0.583
5	0.623
6	0.606
7	0.535
8	0.577
9	0.620
10	0.626
Total	0.590

3. Repeat experiments 1 and 2 for your new k-NN classifier.

Results.

Experiment 1: Old Data

Fold number	Accuracy
1	0.661
2	0.619
3	0.521
4	0.732
5	0.830
6	0.774
7	0.746
8	0.802
9	0.704
10	0.680
Total	0.707

Experiment 2: New Data

Fold number	Accuracy
1	0.676
2	0.661
3	0.718
4	0.619
5	0.633
6	0.563
7	0.563
8	0.633
9	0.746
10	0.520
Total	0.633

4. Now, generate a table of scores (a spreadsheet would work well) with 10-fold scores on the following algorithm variants:

    - k-NN with length normalization
    - k-NN with feature normalization
    - k-NN with length and feature normalization
    - perceptron with length normalization
    - perceptron with feature normalization
    - perceptron with length and feature normalization

    This should be a table with 60 numbers!

Run	k-NN			Perceptron		
	Normalization			Normalization		
	Length	Feature	Both	Length	Feature	Both
1	0.760	0.591	0.605	0.408	0.643	0.639
2	0.732	0.676	0.718	0.591	0.767	0.802
3	0.704	0.845	0.830	0.746	0.816	0.802
4	0.661	0.760	0.746	0.563	0.803	0.811
5	0.647	0.746	0.732	0.633	0.782	0.756
6	0.577	0.746	0.732	0.605	0.790	0.774
7	0.521	0.830	0.830	0.535	0.845	0.837
8	0.732	0.774	0.788	0.591	0.824	0.797
9	0.704	0.746	0.732	0.633	0.804	0.761
10	0.533	0.813	0.813	0.626	0.846	0.798

5. Pick a few (say 4-5) of these results (including the earlier results) and calculate their t-test score to figure out if the differences are significant. Pick a couple of the experimental results that are close and a couple where they're further apart.
    I'd suggest just using Excel/open office to calculate these, though you can use whatever you'd like. If you use these the t-test function is what you want to use. The first two parameters

are the two data sets, the third parameter (tails) should be 2 (two-tailed test) and the fourth parameter (type) should be 1 (paired t-test).
List the comparisons that you made and their t-test p values.

Comparison	p value
k-NN, both normalizers vs. feature normalizer	1
k-NN, both normalizers vs. length normalizer	0.052758937
Perceptron, both normalizers vs. length normalizer	1.72771×10^{-5}
Perceptron, feature normalizer vs. length normalizer	4.46026×10^{-6}
Perceptron, new data vs. old data	0.007976328

6. Write a short (3-4 sentence) paragraph summarizing your results.

Results.