P556 Homework 2

Fall 2020

Due on Oct 14th, 11:59pm

Question 1: Naive Bayes Classifier (40 points)

When a foreigner comes to the States, sometimes they got very confused about baby's names to tell it is a boy's name or a girl's name, like Jack and Jane. For this question, you will implement a Naive Bayes classifier to classify if a given test baby's name is a boy's name or a girl's name. In the dataset, you can find a boy's name list, *boy_names.csv*, which contains 1000 commonly used boy's names. You can also find a girl's name list, *girl_names.csv*, which contains 1000 commonly used girl's names. In order to implement a NB classifier, you need to:

- Compute $P(X|Y)$
- Compute $P(Y)$ but here we don't need you to compute this as indicated in the hint
- Compute log probability $log\frac{P(Y=1|x)}{P(Y=-1|x)}$
- decide the classification result

Hint:

- One critical key here is to convert the strings in names into feature vector. You may implement this as name2vector.py. Think about what are some good features for a name string (i.e. prefixes/suffixes of a boy's name? prefixes/suffixes of a girl's name? Can you hash them?).
- You can assume that the class probability $P(boys) = 0.5$ and also $P(girls) = 0.5$.
- Use a multinomial distribution as model. This will return the probability vectors for all features given a class label.

Questions:

- Implement a NB classifier for the name problem by using the two training files *boy_names.csv* and *girl_names.csv*.
- Test your algorithm using the test dataset *test_names.csv*. For now you can just output your classification result based on which class has higher probability based on your model. Export your result as a .csv file, using "-1" to indicate a boy's name and "+1" to indicate a girl's name.

Question 2: Support Vector Machine (30 points)

2.1 Here is a visualization (Figure 1) of 1200 data points in 2D space ($hw2\_data1.txt$):

Questions:

- Reuse the perceptron you implemented in HW1 question 3, and try to classify the dataset.
- Implement a SVM to classify the dataset.
- Can you tell the resulting difference between these two algorithms? And can you make a plot for the two hyperplanes learned from these two algorithms?

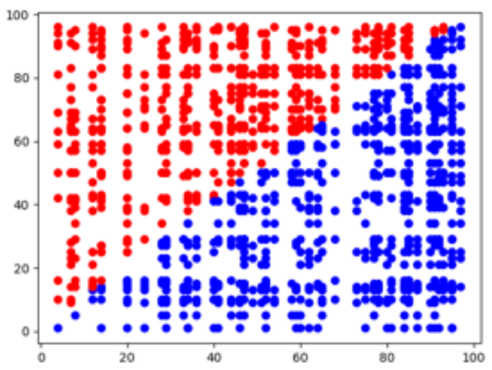

Figure 1: Question 2.1

2.2 Now that you have some hands-on practice synthetic data, here is a real world image classification task called Fashion-MNIST. It is a dataset that consists of 60k training images (*hw2_X_train.gz*) of clothing and shoes, and its labels (*hw2_Y_train.gz*). It also contains a separate set of 10k test images (*hw2_X_test.gz*) and its labels (*hw2_Y_test.gz*). You can read more about the data here `https://github.com/zalandoresearch/fashion-mnist`.

Your goal is to train an SVM using the training set and show your classifier's accuracy on the test set. You can use the SVM you implemented in Question 1. A few things to keep in mind:

- Test data is only to be used for testing. Do not use test data for training. It should be unseen while training.
- Report both accuracies, i.e. the train accuracy and the test accuracy.

Question 3: Naive Bayes Classifier (30 points)

Consider a generative classifier for C classes with class conditional density $p(x|y)$ and uniform class prior $p(y)$. Suppose all the D features are binary, $x_j \in \{0, 1\}$. If we assume all the features are conditionally independent (the naive Bayes assumption), we can write:

$$p(x|y = c) = \prod_{j=1}^{D} Ber(x_j|\theta_{jc})$$

This requires DC parameters.

Questions:

- Generally we assume Bayesian conditionally independence assumption holds. We mean for each dimension we have:
$$P(X = x|Y = y) = \prod_{\alpha} P(x_\alpha|y) \tag{1}$$
Can you describe scenarios when the above equality doesn't hold? Give an example when the left side is bigger than the right side? Give another example when the right side is bigger than the left side?
- Now consider a different model, which we will call the "full" model, in which all the features are fully dependent (i.e., we make no factorization assumptions). How might we represent $p(x|y = c)$ in this case? How many parameters are needed to represent $p(x|y = c)$?
- Assume the number of features D is fixed. Let there be N training cases. If the sample size N is very small, which model (naive Bayes or full) is likely to give lower test set error, and why?