Problem Set 4

CS 6375

Due: 4/21/2022 by 11:59pm

Note: all answers should be accompanied by explanations for full credit. Late homeworks will not be accepted.

Problem 1: PCA and Feature Selection (50pts)

In this problem, we will explore ways that we can use PCA for the problem of generating or selecting "good" features.

SVMs and PCA (25pts)

Consider the prostate_GE data set (attached to this problem set). Each row corresponds to a single data point and the final column in each row is the class label (1 or 2 here). You should use the first 80 data points for training and the remaining 10% for testing.

- Perform PCA on the training data to reduce the dimensionality of the data set (ignoring the class labels for the moment). What are the top six eigenvalues of the data covariance matrix? Looking at the eigenvalues, how would you recommend picking k for this data set?
- For each $k \in \{1, 2, 3, 4, 5, 6\}$, project the training data into the best k dimensional subspace (with respect to the Frobenius norm) and use the SVM with slack formulation to learn a classifier for each $c \in \{1, 10, 100, 1000\}$. As data is limited, this process should be done with 10-fold cross-validation. Report the average error of the learned classifier on the held out validation data for each k and c pair.
- What is the performance you achieve on the test set via the proper hyperparameter selection procedure above?
- Now suppose that we don't do proper hyperparameter selection. What is the best performance that you can achieve on the test set if you tune the hyperparameters using the test set instead of the validation set?

PCA for Feature Selection (25pts)

If we performed PCA directly on the training data as we did in the first part of this question, we would generate new features that are linear combinations of our original features. If instead, we wanted to find a subset of our current features that were good for classification, we could still use PCA, but we would need to be more clever about it. The primary idea in this approach is to select

features from the data that are good at explaining as much of the variance as possible. To do this, we can use the results of PCA as a guide. Implement the following algorithm for a given k and s:

1. Compute the top k eigenvalues and eigenvectors of the covariance matrix corresponding to the data matrix omitting the labels (recall that the columns of the data matrix are the input data points). Denote the top k eigenvectors as $v^{(1)}, \ldots, v^{(k)}$.
2. Define $\pi_j = \frac{1}{k} \sum_{i=1}^{k} {v_j^{(i)}}^2$.
3. Sample s features independently from the probability distribution defined by π.

- Why does π define a probability distribution?
- Again, using the prostate_GE data set and same procedure as above, for each $k \in \{1, 10, 20, 40, 80, 160\}$ with $s = \lfloor k \log k \rfloor$, report the average test error of the SVM with slack classifier over 20 experiments. For each experiment use only the s selected features (note that there may be some duplicates, so only include each feature once). Use the same hyperparameter search for c as in part 1.
- Does this provide a reasonable alternative to the SVM with slack formulation without feature selection on this data set? What are the pros and cons of this approach?

Problem 2: Working with k-means (50pts)

For this problem, you will use the `leaf.data` file provided with this problem set. This data set was generated from the UCI Leaf Data Set (follow the link for information about the format of the data). The class labels are still in the data set and should be used for evaluation only (i.e., don't use them in the clustering procedure), but the specimen number has been removed. You should preprocess the data so that the non-label attributes have mean zero and variance one.

1. Train a k-means classifier for each $k \in \{10, 15, 20, 25, 30\}$ starting from twenty different random initializations (sample uniformly from $[-3, 3]$ for each attribute) for each k. Report the mean and variance of the value of the k-means objective obtained for each k.
2. Random initializations can easily get stuck in suboptimal clusterings. An improvement of the k-means algorithm, known as k-means++, instead chooses an initialization as follows:
   (a) Choose a data point uniformly at random to be the first center.
   (b) Repeat the following until k centers have been selected:
      i. For each data point x compute the distance between x and the nearest cluster center in the current set of centers. Denote this distance as d_x.
      ii. Sample a training data point at random from the distribution p such that $p(x) \propto d_x^2$. Add the sampled point to the current set of centers.

   Repeat the first experiment using this initialization to pick the initial cluster centers for k-means. Does this procedure result in an improvement? Explain.