CS181 Practice Questions: Clustering and PCA

K-Means and Related Algorithms

1. **Convergence of K-Means (Bishop 9.1)**

Consider Lloyd's algorithm for finding a K-Means clustering of N data, i.e., minimizing the "distortion measure" objective function

$$J(\{\boldsymbol{r}_n\}_{n=1}^N, \{\boldsymbol{\mu}_k\}_{k=1}^K) = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} ||\boldsymbol{x}_n - \boldsymbol{\mu}_k||_2^2.$$

Show that as a consequence of there being a finite number of possible assignments for the set of responsibilities $r_{n,k}$, and that for each such assignment there is a unique optimum for the means $\{\boldsymbol{\mu}_k\}_{k=1}^K$, the K-Means algorithm must converge after a finite number of iterations.

Since both the responsibility updates and the mean updates minimize the K-Means objective function, Lloyd's algorithm will never move to a new configuration of responsibility assignments unless the new configuration has a lower objective value. Since there are a finite number of possible assignments, each with a corresponding unique minimum with regard to the $\{\boldsymbol{\mu}_k\}_{k=1}^K$, the K-Means algorithm will converge after a finite number of steps, when no changes to the responsibilities will decrease the objective. When the responsibilities don't change in an iteration, the $\{\boldsymbol{\mu}_k\}_{k=1}^K$ also don't change.

2. **K-Means++**

One way to initialize Lloyd's algorithm for K-Means is to randomly select some of the data to be the first cluster centers. The easiest version of this would pick uniformly from among the data. K-Means++ biases this distribution so that it is not uniform. Explain in words how the distribution is non-uniform and why it should lead to better initializations.

The K-Means++ algorithm iteratively adds cluster centers, drawing them from a distribution over the data. This distribution is proportional to the squared distance of each datum from its nearest cluster center. Thus K-Means++ tends to favor points that are distant from the existing centers and produce a more diverse set of centers.

3. **Standardizing Input Data**

Standardizing data helps ensure that distances makes sense and that the different properties of the items are balanced. Give an example of a kind of data for which standardization might be necessary to get good results from K-Means clustering.

Any example with features that have different units:

- Clustering galaxies by size and brightness.
- Clustering fruit by color and weight.
- Clustering houses by square footage and price.

4. **K-Medoids**

K-Medoids clustering is similar to K-Means, except that it requires the cluster centers to be data examples. Describe a situation in which this is desirable or necessary.

Sometimes we only have distances between points, and averages of points are not sensible or available. In such cases, we cannot describe a cluster by a mean of data and instead describe it by an "exemplar".

Hierarchical Agglomerative Clustering

1. **Curse of Dimensionality**

   Define the concept of "the curse of dimensionality" and explain how it is related to HAC.

   The curse of dimensionality refers to the problem when volume exponentially increases with added dimensions.

   HAC is a nonparametric method that depends on distance to computer clusterings. Unfortunately, the curse of dimensionality means that distances become less meaningful in higher dimensions.

2. **HAC vs K-Means**

What are some advantages of HAC over K-Means?

(a) HAC is deterministic, while K-Means uses a random initialization.

(b) HAC does not require the number of clusters to be pre-specified.

(c) HAC produces hierarchies over data and not just flat clusters.

3. **Single-Linkage HAC**

Using the single-linkage criterion for the HAC algorithm, what is the clustering sequence until there are two clusters remaining? Hint: The single-linkage criterion merges groups based on the shortest distance over all possible pairs.
Step 1: {1} {2} {4} {5} {9} {11} {16} {17}

Step 1: {1} {2} {4} {5} {9} {11} {16} {17}
Step 2-4: {1, 2} {4, 5} {9} {11} {16, 17}
Step 5-6: {1, 2, 4, 5} {9, 11} {16, 17}
Step 7: {1, 2, 4, 5, 9, 11} {16, 17}
Step 8: {1, 2, 4, 5, 9, 11, 16, 17}

Principal Component Analysis

1. **High Dimensional Data (Bishop 12.1.4)**

Suppose we have a design matrix $\boldsymbol{X} \in \mathbb{R}^{N \times D}$ which has been centered, so the sample covariance matrix is $\boldsymbol{S} = \frac{1}{N}\boldsymbol{X}^\top \boldsymbol{X}$. Also, let $\boldsymbol{u}_d$, where $d = 1..D$, be the eigenvectors of $\boldsymbol{S}$.

(a) Show that the D vectors defined by $\boldsymbol{v}_d = \boldsymbol{X}\boldsymbol{u}_d$ are eigenvectors of $\frac{1}{N}\boldsymbol{X}\boldsymbol{X}^\top$, and that they have the same eigenvalues as their corresponding $\boldsymbol{u}_d$.

(b) Assuming we can recover the $\boldsymbol{u}_d$ from the $\boldsymbol{v}_d$ with reasonable time and memory, explain why calculating the $\boldsymbol{v}_d$ first might be useful if $N < D$.

(c) Show that the $\hat{\boldsymbol{u}}_d = \boldsymbol{X}\boldsymbol{v}_d$ is, like $\boldsymbol{u}_d$, an eigenvector of $\boldsymbol{S}$.

(a) We have that $\frac{1}{N}\boldsymbol{X}^\top \boldsymbol{X}\boldsymbol{u}_d = \lambda_d \boldsymbol{u}_d$. Left-multiplying both sides by $\boldsymbol{X}$ gives the result.

(b) This way, we only need to explicitly represent and find the eigenvalues/vectors of an $N \times N$, rather than a $D \times D$ matrix.

(c) From part (a), we have that $\frac{1}{N}\boldsymbol{X}\boldsymbol{X}^\top \boldsymbol{v}_d = \lambda_d \boldsymbol{v}_d$. Left-multiplying both sides by $\boldsymbol{X}^\top$ gives the result. (As Bishop points out, $\hat{\boldsymbol{u}}_d$ is not necessarily normalized).

2. **Heuristic for assessing applicability PCA (Press 9.8, Murphy 12.3)**

Let the empirical covariance matrix $\mathbf{\Sigma}$ have eigenvalues $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_d > 0$. Explain why the variance of the eigenvalues

$$\sigma^2 = \frac{\sum_i^d (\lambda_i - \bar{\lambda})^2}{d},$$

where $\bar{\lambda}$ is the average eigenvalue, is a good measure of whether or not PCA would be useful for analyzing the data.

The higher the value of σ^2 the more useful PCA. If the variance is high, then the values of the eigenvalues drop off quickly, and so the data can be explained with a fewer number of components. However, if the variance is low, a lot of information would be lost from simply keeping a few components, and keeping a large number of components makes the dimensionality reduction useless.

3. **Component vectors**

Suppose I have a dataset with N rows, each row being an instance, and D columns, where each column represents a feature. How many component vectors can we get at most?

We would be able to get at most $\min(N, D)$ component vectors. Usually, we have many more instances than features, so D is the limiting factor. We can keep up to D features, but usually we would use fewer of them to obtain the benefits of PCA. Keeping all D features would retain all information in our original data.