Northeastern University, College of Computer and Information Science

# CS 6220 Data Mining — Assignment 5
# Due Date: October 17th, 2016 at 11:59pm

## K-means Clustering

This assignment will require you to apply and interpret some of the techniques related to K-means clustering that were introduced in class. **An IPython Notebook with predefined functions to assist you with this assignment is available here.** Keep in mind that the main objective of this assignment is to highlight the insights that we can derive from applying these techniques—the coding aspect is secondary. Accordingly, you are welcome to consult any online documentation and/or code that has been posted to the course website, so long as all references and sources are properly cited. You are also encouraged to use code libraries, but be sure to acknowledge any source code that was not written by you by mentioning the original author(s) directly in your submission (comment or header).

**You are expected to submit a single IPython Notebook file following the same instructions and naming convention described in Assignment 0. Answers to the conceptual questions can be embedded in the Notebook as *markdown* cells, and you may use *heading* cells to further organize your document.**

# Assignment Description

**The Data**
For this assignment you will be using the *Iris flower dataset*. The dataset consists of 50 samples from each of three species of Iris, with four features measured from each sample. scikit-learn provides a function to load the dataset (no download required).

**The Idea: Choosing $k$ for $k$-means**
Your objective here will be to assess the performance of $k$-means clustering on the Iris dataset. Recall that the number of clusters, $k$, is an input parameter to the $k$-means algorithm. A variety of measurements are available for estimating the optimal value of $k$. For this assignment, you will look at the sum of squared deviation (SSQ) and the gap statistic. Both of these criteria make use of the intuition that $k$-means tries to minimize variance (the distance or deviation of each point from each of the $k$ clusters) by iteratively assigning points to their nearest clusters.

**Choosing $k$ with SSQ**
The SSQ criterion is a direct application of the intuition that $k$-means tries to minimize variance. Recall that the SSQ criterion sweeps over a range of possible $k$ values, with each value of $k$ associated with a degree of deviation (the distance of each point from each of the $k$ clusters). These deviations can be squared and summed to arrive at the "sum of squared deviation" (SSQ) for each value of $k$. Larger values of $k$ are expected to continue to reduce the SSQ (because there are more clusters for points to be near, reducing their deviation). However, one could expect a leveling-off in the SSQ once the value of $k$ exceeds the true number of clusters, as this would result in true clusters (that is, clusters actually present in the data) being separated. If, then, one plots the SSQ over a range of $k$ values, this leveling-off point may produce a noticeable "elbow" in the plot. By this criterion, the estimated optimal value of $k$ is that which occurs at this elbow point. While simple, the difficulty with this criterion is that often the elbow point is not distinctive or well-defined.

**Choosing $k$ with the Gap Statistic**
The gap statistic provides a criterion that produces a quantifiable estimate of the optimal value of $k$ over a range of possible $k$ values. The intuition here is that there is an expected degree of deviation associated with clustering any given dataset. We want the number of clusters, $k$, that displays the largest "gap" between the deviation we expect, given the dataset and the number of clusters, and the deviation we estimate or observe. Thus, rather than simply considering estimated deviation by itself, we can standardize the estimated deviation for a possible value of $k$ by comparing it with the expected deviation under an appropriate null reference distribution of the data (e.g., a uniform distribution). This difference or gap between the expected deviation and the estimated deviation is termed the gap statistic. Maximizing this gap statistic then corresponds to minimizing the estimated deviation relative to what would be expected. To ensure that we do not needlessly posit additional clusters (i.e., larger values of $k$), we only consider the value $k+1$ if its gap statistic (minus any measurement error) is higher than that for $k$. By this criterion, the lowest value of $k$ with a corresponding gap statistic higher than or equal to the gap statistic of $k+1$ is the estimated optimal value of $k$.

**What to Do**
The provided assignment IPython Notebook includes functions to compute and plot the gap statistic (`gap_statistics` and `plot_gap_statistics`) and the sum of squared deviation (`ssq_statistics` and `plot_ssq_statistics`).

The Iris flower dataset can be loaded in Python by using the following code snippet (with

`datasets` imported from scikit-learn):

```
from sklearn import datasets
# Load the Iris flower dataset
iris = datasets.load_iris()
data = iris.data
```

Then the sum of squared deviations (SSQ) can be easily run and the results plotted by using the following code snippet:

```
# Generate and plot the SSQ statistics
ssqs = ssq_statistics(data, ks=range(1,11+1))
plot_ssq_statistics(ssqs)
```

Similarly, the gap statistic can be easily run and the results plotted by using the following code snippet:

```
# Generate and plot the gap statistics
gaps, errs, difs = gap_statistics(data, nrefs=20, ks=range(1,11+1))
plot_gap_statistics(gaps, errs, difs)
```

Both the SSQ and gap statistic code snippets require a variable `ks`, which defines the range of $k$ values (the "$k$s") to evaluate. For example, if you would like to evaluate $k$ values between 1 an 11 (inclusive), you could set `ks` as `range(1,11+1)`.

The function `plot_gap_statistics` generates two plots. The first plot simply displays the gap statistic for each $k$ value evaluated. The second plot displays the difference between the gap statistic for value $k$ and that computed for value *k+1*. On the second plot, the first non-negative value, or gap difference, is the optimal number of clusters estimated by the gap statistic criterion.

**What to Provide**

Your output should contain the following:

- The SSQs computed for $k$ values between 1 and 10 (inclusive). There should be one plot corresponding to the SSQs.
- The gap statistics computed for $k$ values between 1 and 10 (inclusive). There should be two plots corresponding to the gap statistics.

Given this output, respond to the following questions:

1. Where did you estimate the elbow point to be (between what values of $k$)? What value of $k$ was typically estimated as optimal by the gap statistic? To adequately answer this question, consider generating both measures several times, as there may be some amount of variation in the value of $k$ that they each estimate as optimal.
2. How close are the estimates generated by the elbow point and gap statistic to the number of species of Iris represented in the dataset?
3. Assuming we are trying to generate one cluster for each Iris species represented in the dataset, does one measure seem to be a consistently better criterion for choosing the value of $k$ than the other? Why or why not?