clustering and k-means

Week 7 -- Monday

Clustering = Unsupervised ML Technique

- The difference between supervised and unsupervised machine learning: There are no labels (outputs; target) in the dataset.
- The goal for unsupervised learning is to model the underlying structure or distribution in the data in order to learn more about the data.
- There are several clustering algorithms, including k-means, mean-shift, hierarchical clustering, and DBSCAN
- The various algorithms differ significantly in their notion of what constitutes a cluster and how to efficiently find them (as stated in Wikipedia article here: https://en.wikipedia.org/wiki/Cluster_analysis)

A comparison of the clustering algorithms in scikit-learn

- http://scikit-learn.org/stable/modules/clustering.html

Method name	Parameters	Scalability	Usecase	Geometry (metric used)
K-Means	number of clusters	Very large `n samples`, medium `n_clusters` with MiniBatch code	General-purpose, even cluster size, flat geometry, not too many clusters	Distances between points
Affinity propagation	damping, sample preference	Not scalable with n_samples	Many clusters, uneven cluster size, non-flat geometry	Graph distance (e.g. nearest-neighbor graph)
Mean-shift	bandwidth	Not scalable with `n_samples`	Many clusters, uneven cluster size, non-flat geometry	Distances between points
Spectral clustering	number of clusters	Medium `n samples`, small `n_clusters`	Few clusters, even cluster size, non-flat geometry	Graph distance (e.g. nearest-neighbor graph)
Ward hierarchical clustering	number of clusters	Large `n samples` and `n_clusters`	Many clusters, possibly connectivity constraints	Distances between points
Agglomerative clustering	number of clusters, linkage type, distance	Large `n samples` and `n_clusters`	Many clusters, possibly connectivity constraints, non Euclidean distances	Any pairwise distance
DBSCAN	neighborhood size	Very large `n samples`, medium `n_clusters`	Non-flat geometry, uneven cluster sizes	Distances between nearest points
Gaussian mixtures	many	Not scalable	Flat geometry, good for density estimation	Mahalanobis distances to centers
Birch	branching factor, threshold, optional global clusterer.	Large `n clusters` and `n_samples`	Large dataset, outlier removal, data reduction.	Euclidean distance between points

K-Means clustering

- Works with numeric data only!
- Algorithm:

    1. Pick a number k of random cluster centers
    2. Assign every item to its nearest cluster center using a distance metric
    3. Move each cluster center to the mean of its assigned items
    4. Repeat 2-3 until convergence (change in cluster assignment less than a threshold)

Source: http://www3.nd.edu/~rjohns15/cse40647.sp14/www/content/lectures/12%20-%20k-means%20Clustering.pdf

K-Means clustering: Pros and Cons

• Pros: very efficient (even if multiple runs are performed), can be used for a large variety of data types, easy to understand

• Cons: Susceptible to initialization problems and outliers, restricted to data in which there is a notion of a center, have to specify a k (different starting points may give you different clusters--you won't necessarily get an optimal cluster), requires prior knowledge or assumption about number of clusters

Source: http://www3.nd.edu/~rjohns15/cse40647.sp14/www/content/lectures/12%20-%20k-means%20Clustering.pdf

1) Format and preprocess data for clustering

- Handle missing values (either impute or remove any NaNs)
- Convert any categorical data to numeric (nominal variables need to be dummified)
- Scale data
- Check out this great article on preprocessing data for k-means clustering: http://www.edupristine.com/blog/k-means-algorithm

2) Perform a K-Means clustering analysis --Selecting k

- **Random initialization** – k-Means clustering is prone to initial seeding i.e. random initialization of centroids which is required to kick-off iterative clustering process. Bad initialization may end up getting bad clusters.

So how do we choose the right k?

Source: http://www.edupristine.com/blog/k-means-algorithm

--Selecting k (cont'd)

- Visualize the data for a clue
- Trial and Error (using performance evaluation)
- Domain Knowledge or Literature Review
- Guess

Check out this great article about various methods to select k:

http://www.edupristine.com/blog/beyond-k-means

2) Perform a K-Means clustering analysis

- Set k
- Call instance of algorithm with k
  - kmeans = cluster.KMeans(n_clusters=k)
- Fit model
  - kmeans.fit(dfn)
- Get labels and centroids:
  - labels = kmeans.labels_
  - centroids = kmeans.cluster_centers_

3) Evaluate clusters for fit

- Accuracy Score
- Silhouette Score
- Confusion Matrix
- Classification Report
- Visualization of Results