# COS 424: Interacting with Data

Lecturer: Léon Bottou
Scribe: Zhen James Xiang and Jieqi Yu

Lecture # 7
3/4/2010

## Announcements

To leave more time for the final project, homework 3 will be a continuation of Homework 2 and will be significantly shorter. Also Homework 4 will be focusing on a narrow topic.

## Introduction

To put things in context, we first revisit the "mix and match" slide (slide 2) that describes different aspects of machine learning. For this clustering lecture, we will cover both probabilistic and non-probabilistic representations, and both online and offline algorithms.

What is clustering? From a very high level, clustering means splitting the observations into subsets with similar characteristics. It has many applications (slide 3). The center philosophy is dividing the data into smaller units that are easier to handle. One can then understand the data better by studying both the intra-cluster and inter-cluster relationships.

This lecture is organized by the following 4 topics: (1) What is a cluster? (2) K-Means. (3) Hierarchical Clustering. (4) Simple Gaussian Mixtures.

## What is A Cluster

The example on slide 5 explains the concept of clusters. We have two neatly separated classes (red and blue) in this example. Apparently, if we know the labels of the points then we can easily infer the Bayes decision boundary. But even if we don't know the labels and just observe $\mathbb{P}\{X\}$ (the yellow mass), we can still make sense of the data and infer the decision boundary, because the clusters leave a trace in $\mathbb{P}\{X\}$.

However, is this always the case? Let's consider an input space transformation (slide 6). Under this transformation, each point is mapped to another point and this results in a different input space. In real world the choice of an input space is usually arbitrary because of engineering reasons. For instance, the "pixels" on our retina and the "pixels" on our camera give different input spaces, because in our retina the sensory cells are denser near the fovea and sparser in the peripheral. The engineering (in this case the way that cameras are built) usually dictates us to work in an arbitrary input space.

After such arbitrary input space transformation, we get a different $\mathbb{P}\{X\}$ on slide 7 (the yellow mass). As we can see, although the Bayes decision boundary and classification error rate remain invariant, the neatly separated clusters are gone.

The lesson here is that compared to classification or regression problems, there is usually an additional twist in clustering problems. The clustering result is highly dependent on the input space, which is arbitrary.

## K-Means

The K-Means problem is to minimize the distortion function $C(w)$ defined on slide 8. (In this lecture we use the Euclidean distance as the error measure but other error measures can also be used.) The target function $C(w)$ is not convex. As a matter of fact if $(w_1, w_2, \ldots, w_k)$ is a minimum point, then any permutation of $(w_1, w_2, \ldots, w_k)$ will also be a minimum point.

The problem of minimizing $C(w)$ is NP hard in general. People can contrive examples in which one must examine all the combinations to find a solution. Fortunately this is usually not the case in real world clustering problems.

To solve the K-Means problem, we first introduce the Lloyd's algorithm (slide 9). This is an offline algorithm proposed in the 50's. To initialize the algorithm, we can pick $K$ random observations and use them as the initial centroids. The averaging update

$$w_k \leftarrow \frac{1}{|S_k|} \sum_{i \in S_k} x_i \tag{1}$$

is a result of using Euclidean distance error measure. (If we use the absolute value as the error measure, the updated centroid will be the median rather than the average.)

Slide 10-13 demonstrate the algorithm on a simple data set. Slide 14 explains why Lloyd's algorithm works. The red line stands for $\mathcal{L}(s, w)$, the quantization error. The blue line stands for $C(w)$. The gap between red and blue lines is $\mathcal{D}(s, w)$, which measures the assignment error. Notice that $\mathcal{D}(s, w)$ is always non-negative because $\min_k ||x_i - w_k||^2$ corresponds to the optimal assignment. From the description in step 1 and 2 it is easy to see that Lloyd's algorithm keeps driving down $C(w)$.

Next we introduce an online K-Means algorithm, the MacQueen's algorithm developed by James MacQueen in 1967 (slide 15). Interestingly the term "K-Means" was first coined by James MacQueen in the same paper. MacQueen's algorithm is much faster than the Lloyd's algorithm. One can easily run the algorithm with multiple initializations.

We provide three explanations of why MacQueen's algorithm works (slide 16). In the first explanation we define $u_n$ as the mean of $n$ samples: $u_n = \sum_{i=1}^{n} x_i$. Notice that $u_n$ can be recursively computed by:

$$u_n = u_{n-1} + \frac{1}{n}(x_n - u_{n-1}) \tag{2}$$

The update formulation of MacQueen's algorithm takes exactly the same form:

$$w_{s_t} \leftarrow w_{s_t} + \frac{1}{n_{s_t}}(x_t - w_{s_t}), \tag{3}$$

which makes $w_{s_t}$ to approximate the average $u_n$ of each subset. The second explanation views the update step as a stochastic gradient descent step. Built on this understanding, the third explanation points out that even when considering the second order gradients, the step length $\gamma_t = \frac{1}{n_{s_t}}$ matches the Newton step. Because the Hessian of $C(w)$ is a diagonal matrix with diagonal elements being the fractions of observations assigned to each clusters.

An example application of clustering algorithms is color coding of images. Indexed color image formats (such as gif, tiff) use a small palette of $K$ colors to approximate the general color $(r, g, b)$ in an image. This is a clustering problem and is discussed on slide 17.

One drawback of K-Means algorithms is that we have to specify $K$. There are a lot of literature on how to choose $K$. One interesting method is the elbow method (slide 18). However the method may not always be effective. Because every point on the curve is just a local minimal value. It is possible that these local minimal values will not connect as a smooth curve. One could end up with a zig-zag shaped curve if unlucky.

## Hierarchical Clustering

There are two ways to perform hierarchical clustering (slide 19). One method is agglomerative clustering. In this method, one question is how to decide if two clusters are close.

Different distance measures (minimum distance, maximum distance, average distance) can be used. Another method is divisive clustering, the questions here are how to define the "largest cluster" to divide (is the largest cluster the one with the most points, or the one that can be best divided?) and how to divide it. There are many variants of this method.

In the algorithm on slide 20. We first use a fast online K-Means algorithm (such as MacQueen's algorithm) and then merge the clusters. The derivation uses the equalities:

$$\sum_{x \in A} x = n_A w_A, \quad \sum_{x \in B} x = n_B w_B \tag{4}$$

One can also do a small K-Means in each step instead of merging the clusters. If we do this, the clusters will have a smaller distortion, but they would no longer be hierarchical (i.e. organized as a tree).

The hierarchical clustering algorithm outputs a "dendogram". One example is shown on slide 21. In this example the faculties in a psychology department are hierarchically clustered into a tree structure. The tree is then used to hopefully understand the relationships between these people. This is common in clustering analysis, but one should keep in mind that the clustering result is dependent on the input space and is only a representation of the chosen input space.

### Simple Gaussian Mixtures

Now we turn to the probabilistic formulations. Clustering is essentially a density estimation problem. We mentioned in previous lectures that density estimation is "hopeless" unless we have a parametric model. In this lecture we assume the simple Gaussian mixture model. "Simple" means that the standard deviation is the same for all components and is known. This will keep things simple and more analogous to K-Means. In the next lecture we will discuss mixture models in a more general setting and in more details.

After these assumptions, we can write the maximum likelihood as the last equation on slide 23. It is non convex and suffers from local minimums. One can of course still do maximum likelihood estimation via conjugate gradient descent or Newton method. But there is a simpler method that is very meaningful too.

The main idea of the method is to have a guess for Y, since things will be much simpler if we knew Y. We decompose $\log L(\theta)$ as

$$\log L(\theta) = \mathcal{L}(Q, \theta) - \mathcal{M}(Q, \theta) + \mathcal{D}(Q, \theta) \tag{5}$$

where $\mathcal{L}(Q, \theta), \mathcal{M}(Q, \theta)$ and $\mathcal{D}(Q, \theta)$ are defined on slide 24. Although being a daunting expression, this decomposition allows us to effective maximize $\log L(\theta)$ by optimizing separate terms on the right hand side of (5). In this decomposition, $\mathcal{L}(Q, \theta)$ is a simple Gaussian likelihood and $\mathcal{M}(Q, \theta)$ and $\mathcal{D}(Q, \theta)$ are KL divergence measures.

This results in the EM algorithm (slide 25) that is very similar to the Lloyd's algorithm. The first step is soft assignment (which is also strangely called the E step). The assignment doesn't change $\mathcal{L}(Q, \theta)$. The second step, also called the M step, updates the parameter of Gaussian models, which is easy with fixed assignment.

The major difference between the EM algorithm and the Lloyd's algorithm is that the hard assignment in Lloyd's algorithm is replaced by the soft assignment in EM algorithm. (When parameter $\sigma \to 0$, the soft assignment result converges to the hard assignment result.) As a result , the EM algorithm is more robust to local minima. But as a tradeoff,

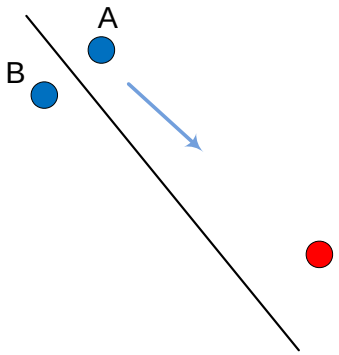


Figure 1: Overlapping Clusters

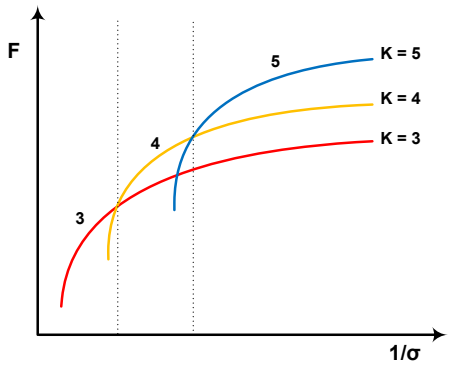


Figure 2: The interaction between $\sigma$ and $K$.

it is slower than the Lloyd's algorithm. The soft assignment also makes the EM algorithm slower in separating different clusters, especially when the clusters have a lot of overlap and $\sigma$ is large. This can be illustrated by the example in Figure 1.

In this example, suppose A and B are two cluster centroids that happen to be close to each other. The red point is a new data point that has almost equal-distance from A and B. In this situation, the EM algorithm may assign a probability of 0.51 that the new data point belongs to A, and a probability of 0.49 that the point belongs to B. Such probability assignment does not affect the position of centroids much in the next step. As a result, the EM algorithm will have a hard time separating these two close-by clusters. On the other hand, the K-Means algorithm has a hard assignment rule and assigns the new point to A. On the next step, the centroid A will move significant toward the new point. This makes the algorithm much faster in separating these two clusters.

The separation of overlapping clusters is controlled by parameters $K$ (the number of clusters) and $\sigma$ (the variance of the simple Gaussian mixture). There is an interesting interplay between these two parameters as shown in Figure 2. In this figure, we plot the following monotonic function $F$ of $L(\theta)$

$$F = \sigma^2 \log L(\theta) + \frac{n}{2}\sigma^2 \log \sigma^2 \tag{6}$$

over $\frac{1}{\sigma}$ for different $K$. The data points are more "willing to separate" under a small $\sigma$ (higher $F$ for larger $K$ when $\frac{1}{\sigma}$ is large).

As an exercise, verify that in Figure 2 the curve $K = 1$ would be a horizontal line.

For the next lecture, we will discuss Expectation Maximization in a general setting. We will talk about the EM algorithm for general Gaussian Mixtures Models and for all kinds of other mixture models, and how to deal with missing data.