The k-means clustering

Jan Šochman

April 30, 2007

1 Introduction

Given a set of vectors $X = \{x_1, \ldots, x_n\}$, the k-means clustering algorithm finds vectors $\mu_1, \ldots, \mu_k$ ($k < n$) such that the mean square distance between X and $\mu_1, \ldots, \mu_k$ is minimal. Informally, k-means algorithm finds k vectors, which well approximate the given dataset, i.e. such vectors, to which the euclidean distance of the given vectors is minimal.

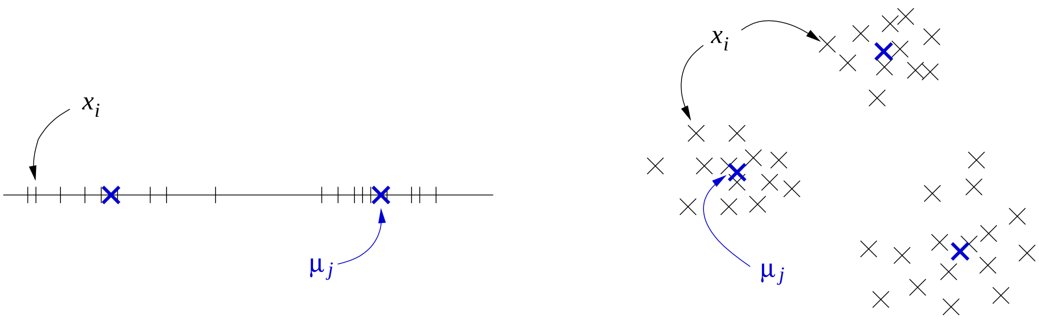


Figure 1: One dimensional (left) and two dimensional (right) example of found vectors $\mu_1, \ldots, \mu_k$.

2 The k-means algorithm

The k-means algorithm is simple. The input consists of a set of vectors $X = \{x_1, \ldots, x_n\}$ and of the number k of sought vectors μ_j.

1. **Initialisation:** Initialise μ_j, $j = 1, \ldots, k$ to random values. Alternatively, heuristics, based on apriori knowledge about a specific task, can be used.

2. **Classification:** Vectors x_i, $i = 1, \ldots, n$ are classified to classes represented by vectors μ_j, $j = 1, \ldots, k$. Each x_i is assigned to the class, which mean vector is the closest (*nearest-neighbour classification*). I.e. x_i is assigned to class

$$y_i = \underset{j=1,\ldots,k}{\operatorname{argmin}} \|x_i - \mu_j\|.$$

3. **Learning:** Update vectors μ_j. μ_j is the mean value of all vectors x_i, which were assigned to j-th class. I.e.

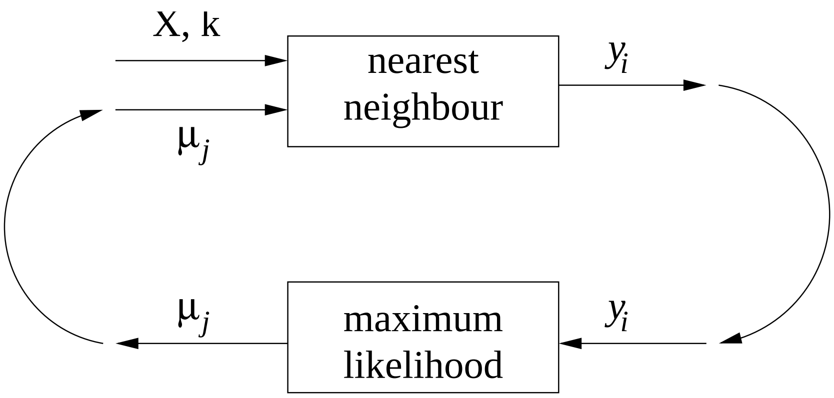
$$\mu_j = \frac{1}{n_j} \sum_{i \in \{i : y_i = j\}} x_i,$$

where n_j is the number of x_is classified to j-th class.

Steps 2 and 3 are iterated as long as the class assignement changes for any x_i.

3 Notes

Look closely at the last step of the algorithm. Observe, that what we compute there is, in fact, the maximum-likelihood estimate of the mean value of each class. The algorithm can therefore be visualised as



We can therefore imagine the data to be drawn from a mixture of several gaussian distributions. Would we assume that all the gaussians have unit variances, the only free parameters that remain are the mean values. The k-means algorithm estimates the means, as well as the 'weights' signifying how much does each of the gaussians contribute to the mixture ($\frac{n_j}{n}$).