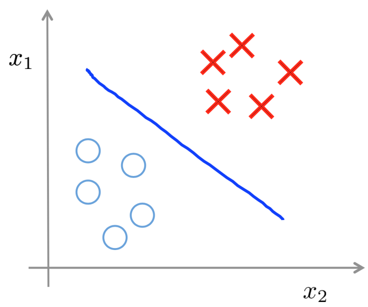
8. Unsupervised Learning: Clustering

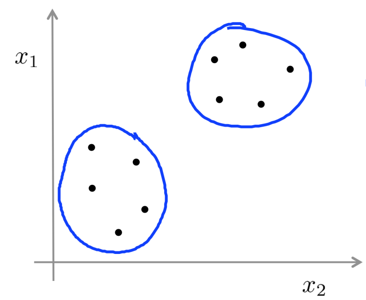
This is our first unsupervised learning algorithm, in which we focus on unlabeled data, rather than labeled data, as shown below:

Supervised Learning



Training set:
$\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(m)}, y^{(m)})\}$

Unsupervised Learning



Training set: $\{x^{(1)}, x^{(2)}, x^{(3)}, \ldots, x^{(m)}\}$

A few key points:

- Notice that our training set no longer has any labels.
- ***Supervised learning:*** Given this dataset, fit a hypothesis to it
- ***Unsupervised learning:*** Ask the algorithm to find structure or a pattern in the dataset.

8.1. k-Means Clustering.

In the clustering problem, we are given a training set $\{x^{(1)}, \ldots, x^{(m)}\}$, and want to group the data into a few cohesive "clusters." Here, $x(i) \in \mathbb{R}^n$ as usual; but no labels $y^{(i)}$ are given. So, this is an ***unsupervised learning*** problem.

The $k-$means clustering algorithm is by far one of the most popular algorithm for clustering. The details are best described by an example:

Example 8.1 : *k-means clustering*

Lets say we are given an unlabeled example dataset and we wish to group the data into two clusters. The k-means clustering algorithm goes as follows:

(a) Initial unlabeled dataset

(b) Randomly initialize the cluster centroids (2-crosses, one for each cluster)

(c) ***Cluster Assignment:*** loops over each example, and, depending on which centroid is closer (red/blue), assigns each datapoint to one of the cluster centroids; or 'paints' each datapoint a color.

(d) ***Centroid Update:*** move cluster centroids to the average position of their own labeled data points; ***e.g.*** compute average of all red points, move red X there.

(e) Repeat Cluster assignment, followed by Move Centroid step.

These steps are highlighted in Fig. 1 below.

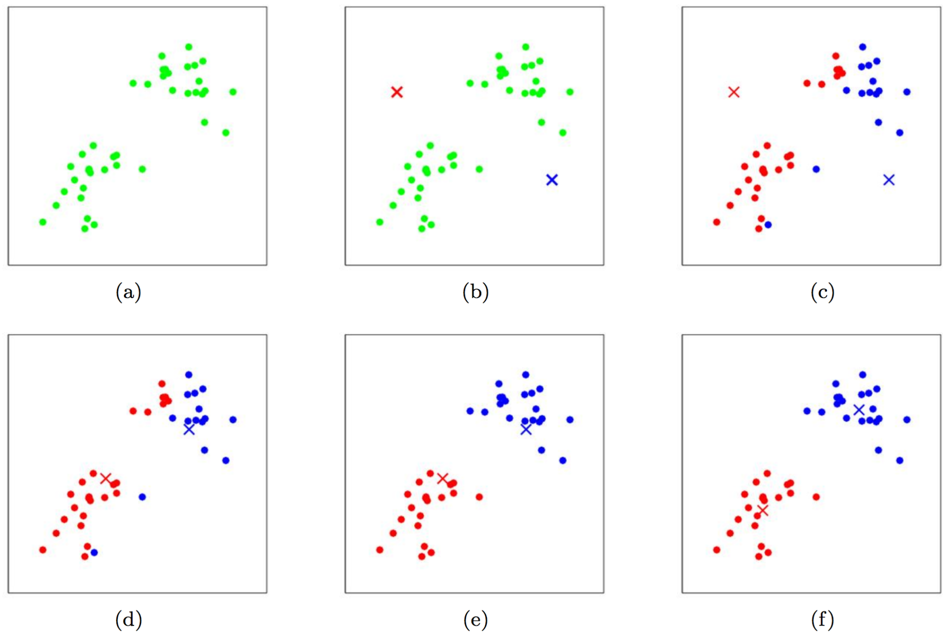


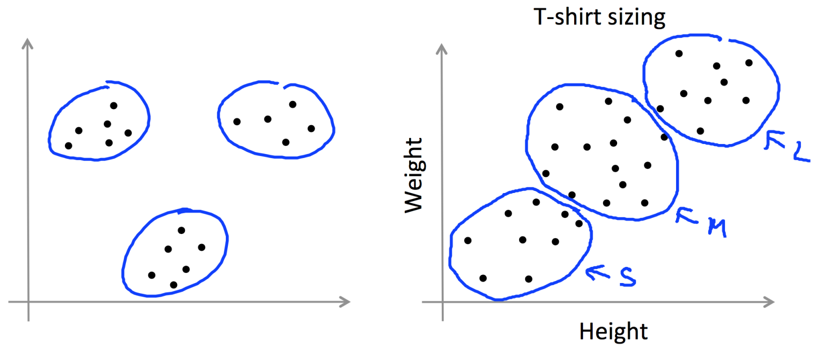
Figure 1. k-means clustering algorithm: Training examples are shown as dots, and cluster centroids are shown as crosses. **(a)** Original dataset. **(b)** Random initial cluster centroids *(in this instance, not chosen to be equal to two training examples)*. **(c-f)** Illustration of running two iterations of k-means. In each iteration, we assign each training example to the closest cluster centroid (shown by "painting" the training examples the same color as the cluster centroid to which is assigned); then we move each cluster centroid to the mean of the points assigned to it. (Best viewed in color.)

Next, writing this algorithm more formally, we have:

Algorithm 1: k-means clustering

1 **Set:** *Initial unlabeled data,* $x^{(i)} \in \mathbb{R}^n$;
2 **Set:** *Cluster Centroids,* $\mu_k \in \mathbb{R}^n$ *(randomly initialized)*;
3 **while** *not converged* **do**
4   ***Cluster Assignment***;
5   **for** $i = 1$ **to** m*-points,* **do**
6         $c^{(i)} := \underset{j}{argmin} \|x^{(i)} - \mu_j\|^2$;
7   **end**
8   ***Centroid Update***;
9   **for** $k = 1$ **to** K, **do**
10         $\mu_j := \frac{\sum_{i=1}^{m} 1\{c^{(i)}=j\} x^{(i)}}{\sum_{i=1}^{m} 1\{c^{(i)}=j\}}$;
11   **end**
12 **end**

In the algorithm above, k (a parameter of the algorithm) is the number of clusters we want to find; and the cluster centroids μ_j represent our current guesses for the positions of the centers of the clusters. To initialize the cluster centroids (in step 1 of the algorithm above), we could choose k training examples randomly, and set the cluster centroids to be equal to the values of these k examples. (Other initialization methods are also possible.)



8.1.1. ***k-means for non-separated clusters*.**

Often enough, the data is not well-separated. The example below shows a t-shirt manufacturing example. The k-means clustering will group the points into a Small, Med, Large, clusters.