1 Clustering

Unlike supervised learning, in which the training examples are (x, y) pairs, the training set of unsupervised learning contains only x without y label. The first type of unsupervised learning problem that we will introduce is clustering. The target of clustering is to group the training examples $\{x^{(1)}, x^{(2)} \dots x^{(m)}\}$ into a few clusters. Clustering is widely applied in scientific research and industrial practice, such as market segmentation, social network analysis, computing clusters organization as well as astronomial data analysis.

1.1 K-means algorithm

The most popular and the most widely used clustering algorithm is K-means. It takes two inputs: the training set $\{x^{(1)}, x^{(2)} \dots x^{(m)}\}$ and the number of expected clusters K. Here training example $x^{(i)} \in \mathbb{R}^n$. By convention, the $x^0 = 1$ that we used in supervised learning is dropped. K-means is an iterative algorithm that can be described as follows.

```
1. K cluster centroids $\mu_1, \mu_2 \dots \mu_K (\in \mathbb{R}^n)$ are randomly initialized.
2. Repeat {
   Cluster assignment:
       for $i = 1 : m$, $c^{(i)} :=$ index of cluster centroid closest to $x^{(i)}$, i.e. $k$
       that minimizes $\|x^{(i)} - \mu_k\|$.
   Move centroids:
       for $k = 1 : K$, $\mu_k :=$ average(mean) of all samples assigned to
       cluster $k$, i.e. $\mu_k := \text{AVG}\{x^{(i)} | c^{(i)} = k\}$.
   }
```

1.2 Optimization objective

As in supervised learning, the optimisation objective of K-means algorithm is the minimization of a cost function:

$$J(c^{(1)}, \dots, c^{(m)}, \mu_1, \dots, \mu_K) = \frac{1}{m} \sum_{i=1}^{m} \|x^{(i)} - \mu_{c^{(i)}}\|^2 \tag{1}$$

This cost function is also called the **distortion function**. Obviously, the cluster assignment process could be interpreted as minimizing $J(c, \mu)$ with respect to c, while the move centroids process could be interpreted as minimizing $J(c, \mu)$ with respect to μ.

1.3 Random initialisation

In the first step of K-means algorithm, the K cluster centroids should be randomly initialized. Usually what we do is to randomly pick K training examples and set μ_k to the values of these training examples. Nonetheless, it is possible that the algorithm may end up with a local optima due to the initial choice of the centroids. In order to combat the problem, we should run K-means several times (50-1000) independently (with different random initialisations), calculate the distortion J for each of the clustering results, and choose the result with the smallest distortion.

1.4 Choice of K

The choice of K could be subtle in a practical clustering problem. "Elbow method" might work in some situations, but more often provides no optimal option. Usually, K-means clustering is run for some later/downstream purpose. The choice of K should aim at better serving the downstream purpose. For instance, if we are running K-means on height/weight data of potential customers of a T-shirt we intend to produce in order to figure out how to segment the customers into groups of different sizes, K should obviously be 3 (S, M, L) or 5(XS, S, M, L, XL), even if the choice of K is completely ambiguous at first sight of the data.

2 Dimensionality reduction

Sometimes we may wish to reduce the dimension of the data for some reason.

Dimensionality reduction could be useful for data compression. If all 3D samples are close to a 2D plane, we can project all samples on this plane, and the 3D data is compressed to 2D. In practice the compression could be huge. For example, in a 8-bit RGB image, each pixel requires 24 bits to store the RGB values, each of them being a 8-bit integer from 0 to 255. If we can cluster the RGB values of all pixels into 16 clusters, which could be carried out with K-means, the RGB values of each pixel could be substituted with the values of the centroid to which it is assigned, and we manage to code the image with 4 bits per pixel plus some overhead (RGB values of the 16 centroids) at the price of the loss of some details. Also, when trying to develop pattern recognization machine learning algorithms such as face detection, the training examples are often of high dimensionality (e.g. 1.6×10^4 dimensions for a 128×128 greyscale image). With proper dimensionality reduction preconditioning of the data, the scale of the data can be significantly reduced (maybe from 10000 dimensions to 100 dimensions), and the algorithm can be significantly accelerated without damaging the ability to solve the problem.

Another situation that calls for dimensionality reduction is when we want to visualize the data. Visualization of the data can sometimes provide some intuitive inspirations on the properties of the data set, but it requires that the data be reduced to 2D or 3D.

Principal component analysis (PCA) is the most popular algorithm for dimensionality reduction.

2.1 Formulation of PCA

Mathematically speaking, we have m n-dimension training examples $x^{(1)}, x^{(2)} \dots x^{(m)} (\in \mathbb{R}^n)$, and we hope to find k directions $u^{(1)}, u^{(2)} \dots u^{(k)} (\in \mathbb{R}^n)$ such that the projections of the training examples $x^{(1)}, x^{(2)} \dots x^{(m)}$ on the subspace spanned by $u^{(1)}, u^{(2)} \dots u^{(k)}$, which are denoted with $z^{(1)}, z^{(2)} \dots z^{(m)} (\in \mathbb{R}^k)$, minimize the projection error

$$Err(U_k) = \frac{1}{m} \sum_{i=1}^{m} \left\| U_k z^{(i)} - x^{(i)} \right\|^2 \tag{2}$$

in which

$$U_k = \left[u^{(1)}, u^{(2)}, \dots, u^{(k)} \right].$$

Note that $z^{(i)}$ could be obtained by

$$z^{(i)} = U_k^{\mathsf{T}} x^{(i)}$$

if we require $u^{(i)}$ to be orthogonal and normalized. Inversely, an approximation of $x^{(i)}$ can be obtained with

$$x_{\text{approx}}^{(i)} = U_k z^{(i)}.$$

2.2 Implementation of PCA

Before applying PCA, the data needs to be preconditioned with data scaling and mean normalization, i.e. x_j should be substituted with $\frac{x_j - \mu_j}{\sigma_j}$, in which μ_j is the mean of x_j and σ_j is its standard deviation.

After the preconditioning of the data, we need to calculate the covariance matrix

$$\Sigma = \frac{1}{m} \sum_{i=1}^{m} x^{(i)} {x^{(i)}}^{\mathsf{T}}. \tag{3}$$

Then we carry out sigular value decomposition of Σ, which is a builtin function of Matlab

```
[U,S,V] = svd(Sigma);
```

We end up with a matrix U that contains all eigen vectors of Σ as its columns. The U_k we want is simply its first k colums.

2.3 Mathematics of SVD

2.4 Choice of k

Typically we choose k to be the smallest value ensuring

$$\frac{\frac{1}{m} \sum_{i=1}^{m} \left\| x_{\text{approx}}^{(i)} - x^{(i)} \right\|^2}{\frac{1}{m} \sum_{i=1}^{m} \left\| x^{(i)} \right\|^2} \leq t \tag{4}$$

in which the threshold t could be 1%, 5%, 10%, etc for different purposes. If $t = 1\%$, we say 99% of variance is retained.

Intuitively, what we should do is to choose the threshold we need, and start from the PCA with $k = 1$. If the final result does not satisfy (4), we go to $k = 2$, and so on. However, the matrix S obtained in `svd` provides us with a better approach. S is a diagonal matrix that satisfies, for a specific k,

$$\frac{\frac{1}{m}\sum_{i=1}^{m}\left\|x_{\text{approx}}^{(i)} - x^{(i)}\right\|^2}{\frac{1}{m}\sum_{i=1}^{m}\left\|x^{(i)}\right\|^2} = 1 - \frac{\sum_{i=1}^{k} S_{ii}}{\sum_{i=1}^{m} S_{ii}}. \tag{5}$$

This significantly simplifies our calculation.

2.5 Good use v.s. bad use

PCA helps to compress the data, which sometimes helps machine learning algorithms to run faster. It is necessary to reduce the data to 2D or 3D when we want to visulize the data. These are good use of PCA.

PCA helps to reduce the dimensionality of the data, and fewer features bring smaller possibility of overfitting. But PCA is not a good way to address overfitting. Regularization is always a better option.

PCA helps to accelerate machine learning algorithms, but sometimes it is unnecessary because the algorithm could have run satisfactorily fast with the raw data. It is always wise and worthwhile to give it a try with the raw data before implementing PCA. If the algorithm runs too slowly or exhausts the storage (memory/disk) so that the task is unlikely to be completed with the raw data, it would be time to turn to PCA.