Lecture 11-12:

Unsupervised learning:

Given a set of unlabelled data $D=\{x_1, x_2, \ldots, x_n\}$, i.e. $n$ points in feature space
Compute:

1. The number of classes
2. Prior probabilities and the class models (we have to provide the families/forms of models)
3. Discovery of new concepts as new classes

We will introduce two popular algorithms
  a). K-mean clustering and its "soft"-version by EM
      --- assuming a Gaussian model for each class.
  b). Mean-shift
      -- Non-parametric seeking the modes in the Parzen window representation.

Rational

The Reasons for unsupervised learning:

1. Collecting and labeling a large data set may be practically infeasible for some applications.
2. Given a small set of labeled samples initially, the system should generalize the models to a large data set, and track the change of the characteristics of the patterns over time.
3. In a general sense, if we treat all input signals from vision, speech, smell, touch, ... as input, then most of the learning problems are unsupervised. The problem becomes learning models for one sense (vision) using other senses (Speech etc), and do it iteratively.
Example 1: Image segmentation

Given an image, we are supposed to partition it into many classes and each class is a “coherent” pattern: Grass, Cheetah, Face, Bull, Ground, in the sense that the pixel intensities fit to a probabilistic model $c=1,2,3,4,5$. Each model represents one type of pattern/concept.

Example 2: Data Clustering

There is no single criterion which is generally applicable. For example, some cluster is round and compact, and the other clusters are elongated and may have holes. Therefore we need a broad range of models.
Example 3: Concept discovery

Sometimes, we need to find the concepts, such as phonemes in a foreign language, or basic elements in an image, through data clustering.
In the following, we find some basic elements (like words) for each type of texture image by K-mean clustering.

Data Clustering

What is data clustering?

Given a set of n unlabeled examples \( D=\{x_1, x_2, \ldots, x_n\} \) in a d-dimensional feature space, we partition the set \( D \) into a number of disjoint subsets

\[
D = \bigcup_{j=1}^{K} D_j \quad \text{and} \quad D_i \cap D_j = \emptyset \quad i \neq j
\]

so that points in each subset are coherent according to certain criterion.
We denote a partition by

\[
\pi = (D_1, D_2, \ldots, D_K)
\]

Thus the problem is formulated as

\[
\pi^* = \arg \min_{\pi} f(\pi)
\]
K-mean clustering

One simple example is the k-mean clustering.

There is an engineering version of K-mean clustering, which is a deterministic algorithm. In a generalized version, it is posed as estimating a mixture density by MLE.

The optimal criterion for K-mean clustering is the total variance of the clusters.

Let $m_1, m_2, \ldots, m_k$ be the mean (centers) of the clusters $D_1, D_2, \ldots, D_k$ respectively, the total variance (TV) is

$$f(\pi) = \sum_{j=1}^{k} \sum_{x \in D_j} (x - m_j)^2$$

K-mean clustering algorithm I

A deterministic version of k-mean clustering is:

1. Initialize a partition $\pi$.
   e.g. randomly choose $K$ points $x_1, x_2, \ldots, x_k$ as centers, for other point $y$, it is put into subset $D_j$, if $\chi$ is the closest center to $y$ among the $K$ centers.
2. Repeat,
   2.1 Compute the mean (mass center) for each cluster $D_j$, $j=1,2,\ldots,K$.
   $$m_j = \frac{1}{|D_j|} \sum_{x \in D_j} x$$
   2.2 For $i=1,2,\ldots,n$ compute $d_i(x) = (x - m_i)^2$ assign $x_i$ to cluster $D_j$ with $j* = \arg\min\{d_1(x), d_2(x), \ldots, d_k(x)\}$
2.3 Exit, if no update.
K-mean clustering algorithm II

A "soft" version of k-mean clustering is to assign a point \( x_i \) to each cluster with a probability \( (p_1, p_2, \ldots, p_K) \) [sum to one]

1. Initialize a partition \( \pi \)
   e.g. randomly choose \( K \) points \( x_1, x_2, \ldots, x_K \) as centers \( m_1, m_2, \ldots, m_K \).
2. Repeat,
   2.1 For \( i = 1, 2, \ldots, n \)
      compute distance from \( x_i \) to cluster center \( m_j \) with \( d_j(x) = (x - m_j)^2 \)
      compute the probability that \( x_i \) belongs to \( D_j \) as \[ p_j(x) = \frac{1}{Z} \exp\left(-\frac{d_j^2}{2\sigma_j^2}\right) \]
   2.2 Compute the mean (mass center) and variance for each cluster \( D_j \), \( j = 1, 2, \ldots, K \).
      \[ m_j = \frac{1}{w_j} \sum_{s \in D_j} p_j(x) x \]
      \[ \sigma_j^2 = \frac{1}{w_j} \sum_{s \in D_j} p_j(x) (x - m_j)^2 \]
   2.3 Exit, if no update.

K-mean clustering algorithm III

In fact, the "soft" version of k-mean clustering is a special case of MLE learning of mixture density

Given : a set of unlabelled training examples \( D = \{x_1, x_2, \ldots, x_n\} \)
Objective : estimate a mixture density

\[
p(x; \theta) = \sum_{j=1}^{K} \sigma_j p_j(x; \theta_j) \quad \sum_{j=1}^{K} \sigma_j = 1
\]

The log-likelihood is

\[
\ell(\theta) = \sum_{i=1}^{n} \log p(x_i; \theta) = \sum_{i=1}^{n} \log \left( \sum_{j=1}^{K} w_j p(x_i; \theta_j) \right)
\]

When the class models are assumed to be Gaussians, then this is the soft K-mean.
The above MLE problem with hidden variables is solved by the Expectation-Maximization (EM) algorithm

Lecture note for Stat 231: Pattern Recognition and Machine Learning
Mean-Shift

A natural extension to the parametric method (EM) is to introduce non-parametric densities and it also assumes unknown number of clusters. This is the idea of Mean-shift algorithm by (Y. Cheng, PAMI August, 1995, D. Comaniciu and Meer 1997)

Suppose we have a non-parametric density function in feature space, each point $x$ follows a Gradient ascent path to its local maximum. Each local maximum and all its domain constitute a cluster.

\[
\hat{f}(x) = \frac{1}{n h^d} \sum_{i=1}^{n} K\left(\frac{x-x_i}{h}\right)
\]

with a kernel function Epanechnikov (minimum mean integrated error)

\[
K_{E}(x) = \begin{cases} \frac{3}{4}(1-x^2) & \text{if } |x| < 1 \\ 0 & \text{otherwise} \end{cases}
\]

\[
\nabla f(x) = \nabla \hat{f}(x) = \frac{1}{n h^d} \sum_{i=1}^{n} \nabla K\left(\frac{x-x_i}{h}\right)
\]

Define mean-shift as

\[
M_n(x) = \frac{1}{n} \sum_{x_i \in S_n(x)} [x_i - x] = \frac{1}{n} \sum_{x_i \in S_n(x)} x_i - x
\]

Then we have

\[
M_n(x) = \frac{h^2}{d+2} \frac{\nabla f(x)}{f(x)}.
\]

The mean shift procedure, obtained by successive
- computation of the mean shift vector $M_n(x)$
- translation of the window $S_n(x)$ by $M_n(x)$. 

Lecture note for Stat 231: Pattern Recognition and Machine Learning
Mean-shift: Example

Figure 4: A $40 \times 20$ window from the image cameraman. (a) Original data (rotated and flipped over for better visualization). (b) Mean shift paths for the points in the central and top (white) plateaus. (c) Filtering result. (d) Segmentation result (see Section 6 for details).

Mean-shift: Example of Segmentation

Lecture note for Stat 231: Pattern Recognition and Machine Learning
Mean-shift: Example

Lecture note for Stat 231: Pattern Recognition and Machine Learning