

NOT PART OF THE FINAL

Unsupervised Learning and Other Neural Networks

- Introduction
- Mixture Densities and Identifiability
- ML Estimates
- Application to Normal Mixtures
- Other Neural Networks

- **Introduction**

- **Previously, all our training samples were labeled: these samples were said “supervised”**
- **We now investigate a number of “unsupervised” procedures which use unlabeled samples**
- **Collecting and Labeling a large set of sample patterns can be costly**
- **We can train with large amounts of (less expensive) unlabeled data, and only then use supervision to label the groupings found, this is appropriate for large “data mining” applications where the contents of a large database are not known beforehand**

- This is also appropriate in many applications when the characteristics of the patterns can change slowly with time (such as in automated food classification as the seasons change)
- Improved performance can be achieved if classifiers running in a unsupervised mode are used
- We can use unsupervised methods to identify features (through clustering) that will then be useful for categorization (or classification)
- We gain some insight into the nature (or structure) of the data

● Mixture Densities and Identifiability

- We shall begin with the assumption that the functional forms for the underlying probability densities are known and that the only thing that must be learned is the value of an unknown parameter vector
- We make the following assumptions:
 - The samples come from a known number c of classes
 - The prior probabilities $P(\omega_j)$ for each class are known ($j = 1, \dots, c$)
 - $P(x | \omega_j, \theta_j)$ ($j = 1, \dots, c$) are known but might be different
 - The values of the c parameter vectors $\theta_1, \theta_2, \dots, \theta_c$ are unknown

- The category labels are unknown

$$P(x | \theta) = \prod_{j=1}^c \overbrace{P(x | \omega_j, \theta_j)}^{\text{component densities}} \cdot \underbrace{P(\omega_j)}_{\text{mixing parameters}}$$

where $\theta = [\theta_1, \theta_2, \dots, \theta_c]'$

- This density function is called a mixture density
- Our goal will be to use samples drawn from this mixture density to estimate the unknown parameter vector θ . Once θ is known, we can decompose the mixture into its components and use a MAP classifier on the derived densities

– Definition

- A density $P(x | \theta)$ is said to be identifiable (or injective) if $\theta \neq \theta'$ implies that there exists an x such that:

$$P(x | \theta) \neq P(x | \theta')$$

As a simple example, consider the case where x is binary and $P(x | \theta)$ is the following mixture:

$$\begin{aligned} P(x | \theta) &= \frac{1}{2} \theta_1^x (1 - \theta_1)^{1-x} + \frac{1}{2} \theta_2^x (1 - \theta_2)^{1-x} \\ &= \begin{cases} \frac{1}{2} (\theta_1 + \theta_2) & \text{if } x = 1 \\ 1 - \frac{1}{2} (\theta_1 + \theta_2) & \text{if } x = 0 \end{cases} \end{aligned}$$

Assume that:

$$P(x = 1 | \theta) = 0.6 \Rightarrow P(x = 0 | \theta) = 0.4$$

by replacing these probabilities values, we obtain:

$$\theta_1 + \theta_2 = 1.2$$

- Thus, we have a case in which the mixture distribution is completely unidentifiable, and therefore unsupervised learning is impossible
- In the discrete distributions, if there are too many components in the mixture, there may be more unknowns than independent equations, and identifiability can become a serious problem!
- While it can be shown that mixtures of normal densities are usually identifiable, the parameters in the simple mixture density

$$P(x | \theta) = \frac{P(\omega_1)}{\sqrt{2\pi}} \exp\left[-\frac{1}{2}(x - \theta_1)^2\right] + \frac{P(\omega_2)}{\sqrt{2\pi}} \exp\left[-\frac{1}{2}(x - \theta_2)^2\right]$$

Cannot be uniquely identified if $P(\omega_1) = P(\omega_2)$
 (we cannot recover a unique θ even from an infinite amount of data!)

- $\theta = (\theta_1, \theta_2)$ and $\theta = (\theta_2, \theta_1)$ are two possible vectors that can be interchanged without affecting $P(x | \theta)$
- Identifiability can be a problem, we always assume that the densities we are dealing with are identifiable!

● ML Estimates

- Suppose that we have a set $D = \{x_1, \dots, x_n\}$ of n unlabeled samples drawn independently from the mixture density

$$p(x | \theta) = \sum_{j=1}^c p(x | \omega_j, \theta_j) P(\omega_j)$$

(θ is fixed but unknown!)

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}} p(D | \theta) \text{ with } p(D | \theta) = \prod_{k=1}^n p(x_k | \theta)$$

The gradient of the log-likelihood is:

$$\nabla_{\theta_i} l = \sum_{k=1}^n P(\omega_i | x_k, \theta) \nabla_{\theta_i} \ln p(x_k | \omega_i, \theta_i)$$

Since the gradient must vanish at the value of θ_i

that maximizes l ($l = \sum_{k=1}^n \ln p(x_k | \theta)$) therefore,

the ML estimate $\hat{\theta}_i$ must satisfy the conditions

$$\sum_{k=1}^n P(\omega_i | x_k, \hat{\theta}) \nabla_{\theta_i} \ln p(x_k | \omega_i, \hat{\theta}_i) = 0 \quad (i = 1, \dots, c)$$

By including the prior probabilities as unknown variables, we finally can show that:

$$\hat{P}(\omega_i) = \frac{1}{n} \sum \hat{P}(\omega_i | x_k, \hat{\theta})$$

$$\text{and } \sum_{k=1}^n \hat{P}(\omega_i | x_k, \hat{\theta}) \nabla_{\theta_i} \ln p(x_k | \omega_i, \hat{\theta}_i) = 0$$

$$\text{where : } \hat{P}(\omega_i | x_k, \hat{\theta}) = \frac{p(x_k | \omega_i, \hat{\theta}_i) \hat{P}(\omega_i)}{\sum_{j=1}^c p(x_k | \omega_j, \hat{\theta}_j) \hat{P}(\omega_j)} \quad \text{This equation enables clustering}$$

● Applications to Normal Mixtures

$$p(\mathbf{x} \mid \omega_i, \theta_i) \sim N(\mu_i, \Sigma_i)$$

| Case | μ_i | Σ_i | $P(\omega_i)$ | c |
|------|---------|--------------|---------------|--------------|
| 1 | ? | \mathbf{x} | \mathbf{x} | \mathbf{x} |
| 2 | ? | ? | ? | \mathbf{x} |
| 3 | ? | ? | ? | ? |

Case 1 = Simplest case

– Case 1: Unknown mean vectors

$$\mu_i = \theta_i \quad \forall i = 1, \dots, c$$

$$\ln p(\mathbf{x} \mid \omega_i, \mu_i) = -\ln \left[(2\pi)^{d/2} |\Sigma_i|^{1/2} \right] - \frac{1}{2} (\mathbf{x} - \mu_i)^t \Sigma_i^{-1} (\mathbf{x} - \mu_i)$$

ML estimate of $\mu = (\mu_i)$ is:

$$\hat{\mu}_i = \frac{\sum_{k=1}^n P(\omega_i \mid \mathbf{x}_k, \hat{\mu}) \mathbf{x}_k}{\sum_{k=1}^n P(\omega_i \mid \mathbf{x}_k, \hat{\mu})} \quad (1)$$

Weighted average of the samples
Coming from the i-th class

$P(\omega_i \mid \mathbf{x}_k, \hat{\mu})$ is the fraction of those samples

having value \mathbf{x}_k that come from the i-th class, and $\hat{\mu}_i$ is the average of the samples coming from the i-th class.

- Unfortunately, equation (1) does not give $\hat{\mu}_i$ explicitly
- However, if we have some way of obtaining good initial estimates $\hat{\mu}_i(0)$ for the unknown means, therefore equation (1) can be seen as an iterative process for improving the estimates

$$\hat{\mu}_i(j+1) = \frac{\sum_{k=1}^n P(\omega_i | \mathbf{x}_k, \hat{\mu}(j)) \mathbf{x}_k}{\sum_{k=1}^n P(\omega_i | \mathbf{x}_k, \hat{\mu}(j))}$$

- This is a gradient ascent for maximizing the log-likelihood function
- Example (Class):
Consider the simple two-component one-dimensional normal mixture

$$p(x | \mu_1, \mu_2) = \frac{1}{3\sqrt{2\pi}} \exp\left[-\frac{1}{2}(x - \mu_1)^2\right] + \frac{2}{3\sqrt{2\pi}} \exp\left[-\frac{1}{2}(x - \mu_2)^2\right]$$

(2 clusters!)

Let's draw $n=25$ samples sequentially from this mixture (see Table p.524: \mathbf{x}_k with ω_k) with $\mu_1 = -2$, and $\mu_2 = 2$ The log-likelihood function is:

$$l(\mu_1, \mu_2) = \sum_{k=1}^n \ln p(\mathbf{x}_k | \mu_1, \mu_2)$$

The maximum value of l occurs at:

$$\hat{\mu}_1 = -2.130 \text{ and } \hat{\mu}_2 = 1.668$$

(which are not far from the true values: $\mu_1 = -2$ and $\mu_2 = +2$)

There is another peak at $\hat{\mu}_1 = 2.085$ and $\hat{\mu}_2 = -1.257$ which has almost the same height as can be seen from the following figure:

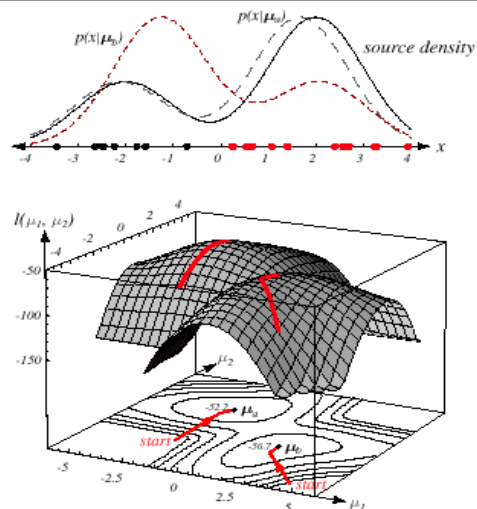


FIGURE 10.1. (Above) The source mixture density used to generate sample data, and two maximum-likelihood estimates based on the data in the table. (Bottom) Log-likelihood of a mixture model consisting of two univariate Gaussians as a function of their means, for the data in the table. Trajectories for the iterative maximum-likelihood estimation of the means of a two-Gaussian mixture model based on the data are shown as red lines. Two local optima (with log-likelihoods -52.2 and -56.7) correspond to the two density estimates shown above. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

- This mixture of normal densities is identifiable
- When the mixture density is not identifiable, the ML solution is not unique

– Case 2: All parameters unknown

- No constraints are placed on the covariance matrix

Let $p(x | \mu, \sigma^2)$ be the two-component normal mixture:

$$p(x | \mu, \sigma^2) = \frac{1}{2\sqrt{2\pi}\sigma} \exp\left[-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right] + \frac{1}{2\sqrt{2\pi}} \exp\left[-\frac{1}{2}x^2\right]$$

Suppose $\mu = x_1$, therefore:

$$p(x_1 | \mu, \sigma^2) \geq \frac{1}{2\sqrt{2\pi}\sigma} + \frac{1}{2\sqrt{2\pi}} \exp\left[-\frac{1}{2}x_1^2\right]$$

For the rest of the samples:

$$p(x_k | \mu, \sigma^2) \geq \frac{1}{2\sqrt{2\pi}} \exp\left[-\frac{1}{2}x_k^2\right]$$

Finally,

$$p(x_1, \dots, x_n | \mu, \sigma^2) \geq \underbrace{\left\{ \frac{1}{\sigma} + \exp\left[-\frac{1}{2}x_1^2\right] \right\}}_{\substack{\text{(this term } \rightarrow \infty \\ \sigma \rightarrow 0)}} (2\sqrt{2\pi})^n \exp\left[-\frac{1}{2}\sum_{k=2}^n x_k^2\right]$$

The likelihood is therefore large and the maximum-likelihood solution becomes singular.

• **Adding an assumption**

Consider the largest of the finite local maxima of the likelihood function and use the ML estimation.

We obtain the following:

Iterative scheme

$$\hat{\mathbf{P}}(\omega_i) = \frac{1}{n} \sum_{k=1}^n \hat{\mathbf{P}}(\omega_i | \mathbf{x}_k, \hat{\theta})$$

$$\hat{\mu}_i = \frac{\sum_{k=1}^n \hat{\mathbf{P}}(\omega_i | \mathbf{x}_k, \hat{\theta}) \mathbf{x}_k}{\sum_{k=1}^n \hat{\mathbf{P}}(\omega_i | \mathbf{x}_k, \hat{\theta})}$$

$$\hat{\Sigma}_i = \frac{\sum_{k=1}^n \hat{\mathbf{P}}(\omega_i | \mathbf{x}_k, \hat{\theta}) (\mathbf{x}_k - \hat{\mu}_i)(\mathbf{x}_k - \hat{\mu}_i)^t}{\sum_{k=1}^n \hat{\mathbf{P}}(\omega_i | \mathbf{x}_k, \hat{\theta})}$$

Where:

$$\hat{\mathbf{P}}(\omega_i | \mathbf{x}_k, \hat{\theta}) = \frac{|\Sigma_i|^{-1/2} \exp\left[-\frac{1}{2}(\mathbf{x}_k - \hat{\mu}_i)^t \hat{\Sigma}_i^{-1}(\mathbf{x}_k - \hat{\mu}_i)\right] \hat{\mathbf{P}}(\omega_i)}{\sum_{j=1}^c |\hat{\Sigma}_j|^{-1/2} \exp\left[-\frac{1}{2}(\mathbf{x}_k - \hat{\mu}_j)^t \hat{\Sigma}_j^{-1}(\mathbf{x}_k - \hat{\mu}_j)\right] \hat{\mathbf{P}}(\omega_j)}$$

– K-Means Clustering

- **Goal:** find the c mean vectors $\mu_1, \mu_2, \dots, \mu_c$
- **Replace the squared Mahalanobis distance**

$(\mathbf{x}_k - \hat{\mu}_i)^t \hat{\Sigma}_i^{-1}(\mathbf{x}_k - \hat{\mu}_i)$ by the squared Euclidean distance $\|\mathbf{x}_k - \hat{\mu}_i\|^2$

- Find the mean $\hat{\mu}_m$ nearest to \mathbf{x}_k and approximate $\hat{\mathbf{P}}(\omega_i | \mathbf{x}_k, \hat{\theta})$ as: $\hat{\mathbf{P}}(\omega_i | \mathbf{x}_k, \hat{\theta}) \cong \begin{cases} 1 & \text{if } i = m \\ 0 & \text{otherwise} \end{cases}$

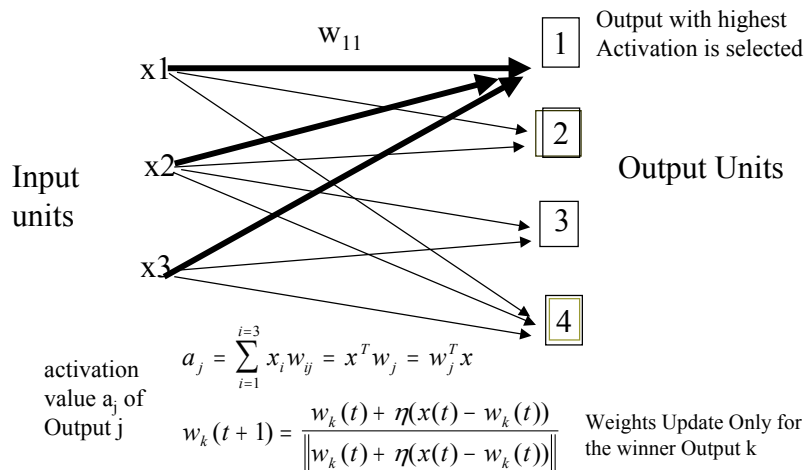
- Use the iterative scheme to find $\hat{\mu}_1, \hat{\mu}_2, \dots, \hat{\mu}_c$
- if n is the known number of patterns and c the desired number of clusters, the k-means algorithm is:

```

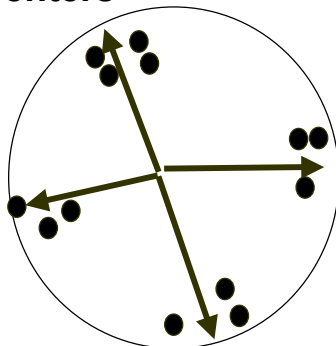
Begin
  initialize  $n, c, \mu_1, \mu_2, \dots, \mu_c$  (randomly
  selected)
  do classify  $n$  samples according to
    nearest  $\mu_i$ 
    recompute  $\mu_i$ 
  until no change in  $\mu_i$ 
  return  $\mu_1, \mu_2, \dots, \mu_c$ 
End
    
```

Other Neural Networks:

- **Competitive Learning Networks (winner-take-all)**



- **Weight vectors move towards those area where most input appears**
- **The weight vectors become the cluster centers**



The weight update finds the cluster centers

The following topics can be considered by the students for their oral presentations

- **Kohonen Self-Organizing Networks**
- **Learning Vector Quantization**