## The Bayes Classifier

- **Maximum likelihood methods assume the parameter vector is unknown but do not assume its values are random. Prior knowledge of possible parameter values can be used to initialize iterative learning procedures.**
- **The Bayesian approach to unsupervised learning assumes $\boldsymbol{\theta}$ is a random variable with known prior distribution $p(\boldsymbol{\theta})$, and uses the training samples to compute the posterior density $p(\boldsymbol{\theta}|D)$.**
- **We assume:**
  - **The number of classes, $c$, is known.**
  - **The prior probabilities, $P(\omega_j)$, for each class are known, $j = 1, \ldots, c$.**
  - **The forms of the class-conditional probability densities, $p(\mathbf{x}|\omega_j,\boldsymbol{\theta}_j)$ *are known*, $j = 1, \ldots, c$, but the full parameter vector $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_c)^t$ is unknown.**
  - **Part of our knowledge about $\boldsymbol{\theta}$ is contained in a set $D$ of $n$ samples $\mathbf{x}_1, \ldots, \mathbf{x}_n$ drawn independently from:**

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

- **We can write the posterior for the class assignment as a function of the feature vector and samples:**

$$P(\omega_i|\mathbf{x},D) = \frac{p(x|\omega_i,D)P(\omega_i|D)}{\sum_{j=1}^{c} p(\mathbf{x}|\omega_j,D)P(\omega_j|D)}$$

## The Bayes Classifier (cont.)

- Because the state of nature, $\omega_i$ , is independent of the previously drawn samples, $P(\omega_i|D) = P(\omega_i)$, we obtain:

$$P(\omega_i \mid \mathbf{x}, D) = \frac{p(\mathbf{x} \mid \omega_i, D)P(\omega_i)}{\sum_{j=1}^{c} p(\mathbf{x} \mid \omega_j, D)P(\omega_j)}$$

- We can introduce a dependence on the parameter vector, $\boldsymbol{\theta}$ :

$$\begin{aligned} p(\mathbf{x} \mid \omega_i, D) &= \int p(\mathbf{x}, \boldsymbol{\theta} \mid \omega_i, D)d\boldsymbol{\theta} \\ &= \int p(\mathbf{x}, \boldsymbol{\theta}, \omega_i, D)\, p(\boldsymbol{\theta} \mid \omega_i, D)d\boldsymbol{\theta} \end{aligned}$$

- Selection of x is independent of the samples: $p(\mathbf{x} \mid \boldsymbol{\theta}, \omega_i, D) = p(\mathbf{x} \mid \boldsymbol{\theta}_i, \omega_i)$

- Because the class assignment when x is selected tells us nothing about the distribution of $\boldsymbol{\theta}$ : $p(\boldsymbol{\theta} \mid \omega_i, D) = p(\boldsymbol{\theta} \mid D)$

- We can now write a simplified expression for $p(\mathbf{x} \mid \omega_i, D)$:

$$p(\mathbf{x} \mid \omega_i, D) = \int p(\mathbf{x} \mid \boldsymbol{\theta}_i, \omega_i)\, p(\boldsymbol{\theta} \mid D)d\boldsymbol{\theta}$$

- Our best estimate of $p(\mathbf{x} \mid \omega_i)$ is obtained by averaging $p(\mathbf{x} \mid \boldsymbol{\theta}_i, \omega_i)$ over $\boldsymbol{\theta}_i$. The accuracy of this estimate depends on our ability to estimate $p(\boldsymbol{\theta} \mid D)$.

## Learning The Parameter Vector

- **Using Bayes formula we can write an expression for $p(\boldsymbol{\theta} \mid D)$:**

$$P(\boldsymbol{\theta} \mid D) = \frac{p(D \mid \boldsymbol{\theta})\, p(\boldsymbol{\theta})}{\int p(D \mid \boldsymbol{\theta})\, p(\boldsymbol{\theta})\, d\boldsymbol{\theta}}$$

- **We can write an expression for $p(D \mid \boldsymbol{\theta})$ using our independence assumption (see slide 11, lecture no. 7):**

$$p(\boldsymbol{\theta} \mid D^n) = \frac{p(\mathbf{x}_n \mid \boldsymbol{\theta})\, p\left(\boldsymbol{\theta} \mid D^{n-1}\right)}{\int p(\mathbf{x}_n \mid \boldsymbol{\theta})\, p\left(\boldsymbol{\theta} \mid D^{n-1}\right) d\boldsymbol{\theta}}$$

- **If $p(\boldsymbol{\theta})$ is uniform over the region where $p(D \mid \boldsymbol{\theta})$ peaks, then (from Bayes rule), $p(\boldsymbol{\theta} \mid D)$ peaks at the same location. If the only significant peak occurs at $\boldsymbol{\theta} = \hat{\boldsymbol{\theta}}$ and if the peak is very sharp, then:**

$$p(\mathbf{x} \mid \omega_i, D) \cong p(\mathbf{x} \mid \omega_i, \hat{\boldsymbol{\theta}})$$

**and**

$$P(\omega_i \mid \mathbf{x}, D) \cong \frac{p(\mathbf{x} \mid \omega_i, \hat{\boldsymbol{\theta}}_i) P(\omega_i)}{\sum_{j=1}^{c} p(\mathbf{x} \mid \omega_j, \hat{\boldsymbol{\theta}}_j) P(\omega_j)}$$

- **This is our justification for using the maximum likelihood estimate, $\hat{\boldsymbol{\theta}}$, as if it were the true value of $\boldsymbol{\theta}$ in designing the Bayes classifier.**
- **In the limit of large amounts of data, the Bayes and ML estimates will be very close.**

## Additional Considerations

- **If $p(\boldsymbol{\theta})$ has been obtained by supervised learning from a large set of labeled samples, it will be far from uniform and it will have a dominant influence on $p(\boldsymbol{\theta}|D^n)$ when $n$ is small.**
- **Each sample sharpens $p(\boldsymbol{\theta}|D^n)$. In the limit it will converge to a Dirac delta function centered at the true value of $\boldsymbol{\theta}$ .**
- **Thus, even though we don't know the categories of the samples, identifiabiliy assures us that we can learn the unknown parameter $\boldsymbol{\theta}$.**
- **Unsupervised learning of parameters is very similar to supervised learning.**
- **One significant difference: with supervised learning the lack of identifiability means that instead of obtaining a unique parameter vector, we obtain an equivalent class of parameter vectors.**
- **For unsupervised training, a lack of identifiability means that even though $p(\boldsymbol{\theta}|D^n)$ might converge to $p(\mathbf{x})$, $p(\mathbf{x}|\theta_i,D^n)$ will not in general converge to $p(\mathbf{x}|\theta_i)$. In such cases a few labeled training samples can have a big impact on your ability to decompose the mixture distribution into its components.**

## Decision-Directed Approximation

- Because the difference between supervised and unsupervised learning is the presence of labels, it is natural to propose the following:
  - Use prior information to train a classifier.
  - Label new data with this classifier.
  - Use the new labeled samples to train a new (supervised) classifier.
- This approach is known as the decision-directed approach to unsupervised learning.
- Obvious dangers include:
  - If the initial classifier is not reasonably good, the process can diverge.
  - The tails of the distribution tend not to be modeled well this way, which results in significant overlap between the component densities.
- In practice, this approach works well because it is easy to leverage previous work for the initial classifier.
- Also, it is less computationally expensive than the pure Bayesian unsupervised learning approach.

## Similarity Measures

- How should we measure similarity between samples?
- How should we evaluate a partitioning of a set of samples into clusters?
- The answer to both requires an ability to measure similarity in a way that is meaningful to the problem. For example, in comparing two spectra of a signal for an audio application, we would hope a spectral distance measure compares closely to human perception. We often refer to this as a "perceptually-meaningful" distance measure, and this concept is more general than audio.
- Principal components can also be used to achieve invariance prior to clustering through normalization.
- One broad class of metrics is the Minkowski metric:

$$d(\mathbf{x},\mathbf{x}') = \left( \sum_{k=1}^{d} \left| x_k - x'_k \right|^q \right)^{1/q}$$

  - Another nonmetric approach measures the angle between two vectors:

$$s(\mathbf{x},\mathbf{x}') = \frac{\mathbf{x}^t \mathbf{x}'}{\|\mathbf{x}\| \, \|\mathbf{x}'\|}$$

## Criterion Functions For Clustering

- **Sum of squared errors:**

$$\mathbf{m}_i = \frac{1}{n_i} \sum_{x \in D_i} \mathbf{x} \qquad J_e = \sum_{i=1}^{c} \sum_{x \in D_i} \left\| \mathbf{x} - \mathbf{m}_i \right\|^2$$

- **Minimum variance criteria:**

$$J_e = \frac{1}{2} \sum_{i=1}^{c} n_i \bar{s}_i \qquad \bar{s}_i = \frac{1}{n_i^2} \sum_{\mathbf{x} \in D_i} \sum_{\mathbf{x}' \in D_i} s(\mathbf{x}, \mathbf{x}')$$

- **Scatter Matrices (e.g., $S_T = S_W + S_B$)**
- **Trace Criterion:**

$$tr[\mathbf{S}_W] = \sum_{i=1}^{c} \sum_{\mathbf{x} \in D_i} \left\| \mathbf{x} - \mathbf{m}_i \right\|^2 = J_e$$

- **Determinant Criterion:**

$$J_d = \left| \mathbf{S}_W \right| = \left| \sum_{i=1}^{c} \mathbf{S}_i \right|$$

- **Invariant Criteria (eigenvalues are invariant under linear transformations):**

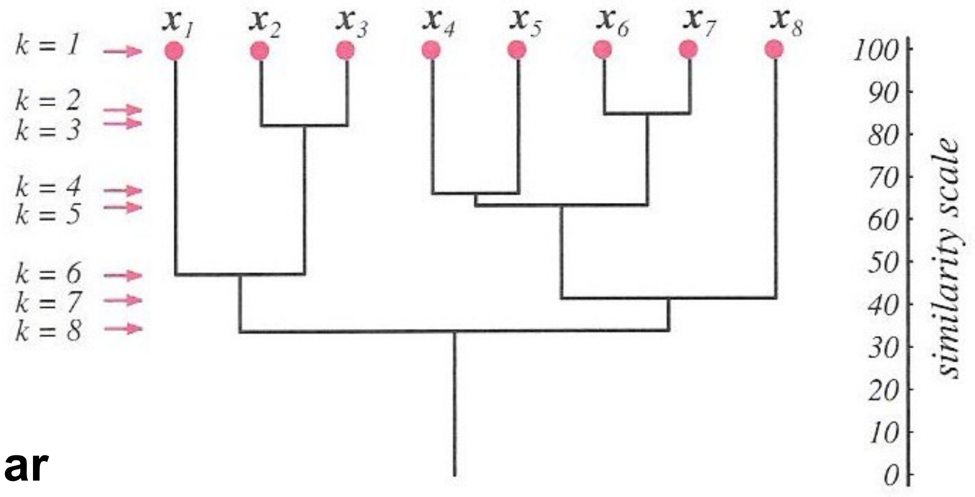
$$tr\left[\mathbf{S}_W^{-1} \mathbf{S}_B\right] = \sum_{i=1}^{d} \lambda_i \qquad J_f = tr\left[\mathbf{S}_T^{-1} \mathbf{S}_W\right] = \sum_{i=1}^{d} \frac{1}{1+\lambda_i} \qquad \frac{\left|\mathbf{S}_W\right|}{\left|\mathbf{S}_T\right|} = \prod_{i=1}^{d} \frac{1}{1+\lambda_i}$$

## Iterative Optimization

- Because the sample set is finite, clustering can be viewed as a problem solved by exhaustive enumeration. However the computational complexity of this approach is prohibitive ($c^n/c!$).
- However, we can apply an iterative hill-climbing procedure:
  - Generate an initial clustering.
  - Randomly select a point.
  - Test whether assigning it to another cluster will reduce the error (e.g., mean squared error); move to the appropriate cluster.
  - Iterate until no further reassignments are made.
- Note that efficient methods exist for updating the cluster means and overall error because we only need to consider the contribution from this one point.
- This can be considered a sequential form of the k-Means algorithm.
- Obvious drawbacks include the potential to get stuck in local minima. One way around this is to use a hybrid approach (e.g., alternate between this and standard *k*-Means algorithm).

## Hierarchical Clustering

- We seek a clustering technique that imposes a hierarchical structure on the data, much like a decision tree.
- Let us consider a sequence of partitions of *n* samples into *c* clusters.
- In the first partition, there are *n* clusters, each with one sample.
- In the next partition, there are *n-1* clusters. At level *k*, there are *n-k+1* clusters.
- If two samples in a cluster at level k remain in the same cluster for higher levels, the clustering technique is referred to as hierarchical clustering.
- This graphical representation of this process shown to the right is referred to as a dendrogram.
- Similarity values can be used to help determine whether groupings are natural or forced (e.g., if the values are comparable across a level, then there is probably no strong argument for any particular clustering of the data.

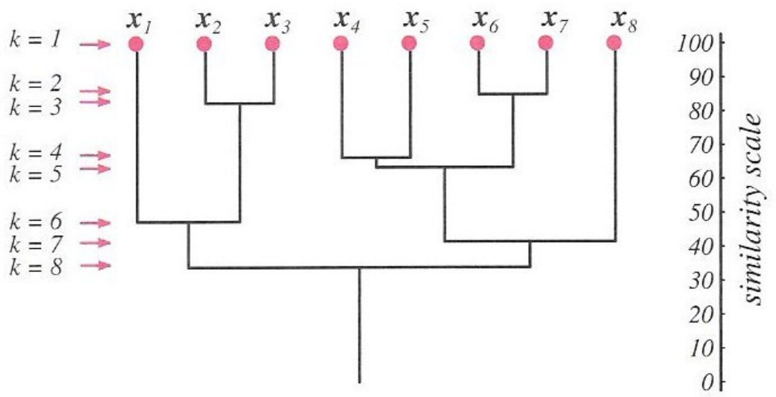


- Venn diagrams can also be used to depict relationships between clusters.

# Agglomerative Clustering

- Hierarchical clustering is very popular as an unsupervised clustering method.
- Two distinct approaches: (1) agglomerative – bottom up, and (2) divisive (top down). The well-known Linde-Buzo-Gray (LBG) algorithm is divisive.
  - Agglomerative clustering typically requires less computation to go from one level to the next.
  - Divisive clustering requires less computation if the goal is a small number of clusters (e.g., c/n << 1).
- Agglomerative hierarchical clustering:

```
▪ Begin: initialize $c, \hat{c} \leftarrow n, D_i \leftarrow \{\mathbf{x}_i\}, i = 1, \ldots, n$
    ○ Do $\hat{c} \leftarrow \hat{c} - 1$
        ➢ Find nearest clusters, $D_i$ and $D_j$
        ➢ Merge $D_i$ and $D_j$
    ○ Until $c = \hat{c}$
▪ Return $c$ clusters
▪ End
```



- If we continue this process until *c = 1*, we produce the dendrogram shown above.

## Agglomerative Clustering (cont.)

- **It is common to use minimum variance type distance measures:**

$$d_{\min}(D_i, D_j) = \min_{\substack{\mathbf{x} \in D_i \\ \mathbf{x}' \in D_i}} \|\mathbf{x} - \mathbf{x}'\|$$

$$d_{\max}(D_i, D_j) = \max_{\substack{\mathbf{x} \in D_i \\ \mathbf{x}' \in D_i}} \|\mathbf{x} - \mathbf{x}'\|$$

$$d_{avg}(D_i, D_j) = \frac{1}{n_i n_j} \sum_{\mathbf{x} \in D_i} \sum_{\mathbf{x}' \in D_i} \|\mathbf{x} - \mathbf{x}'\|$$

$$d_{mean}(D_i, D_j) = \|\mathbf{m}_i - \mathbf{m}_j\|$$

- **When d$_{min}$ is used, this is referred to as a nearest-neighbor cluster algorithm.**
- **Computational complexity:**
  - **Need to calculate** $n(n\text{-}1)$ **interpoint distances, each is** $O(d)$**, and store them in a table, which is** $O(n^2)$ **in memory.**
  - **Finding the minimum distance pair requires stepping through the entire list.**
  - **Overall complexity:** $O(n(n\text{-}1)(d+1)) = O(n^2 d)$. **In general,** $O(cn^2 d)$ **where** $n >> c$.
- **But there are faster implementations of these approaches. Memory becomes a major bottleneck for large data sets. Disk caching becomes important.**

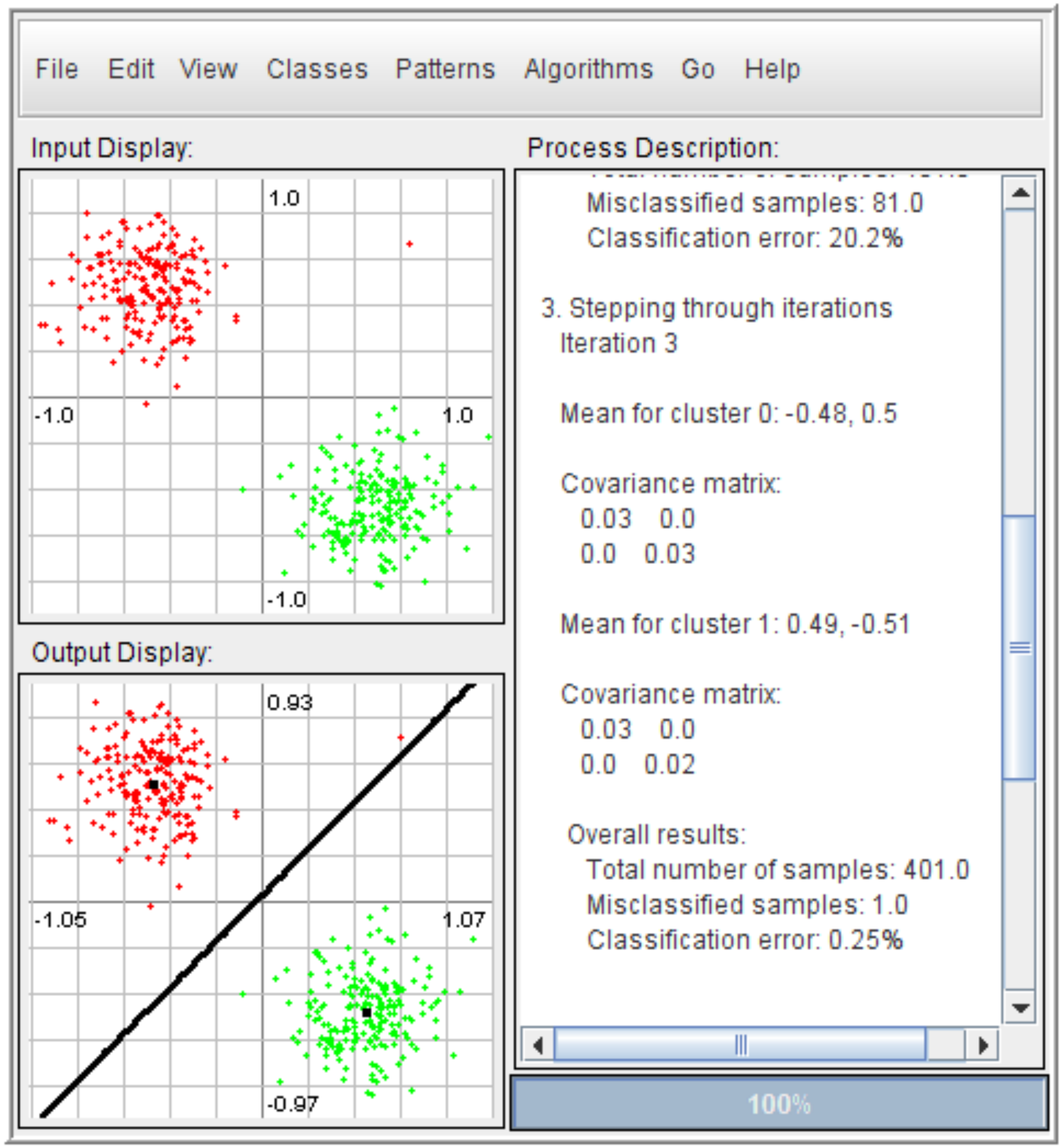
## Spanning Trees

- **Single-linkage algorithm**: The clustering algorithm is terminated when the distance between the nearest clusters exceeds an arbitrary threshold.
- **Spanning Tree:** A tree with any path from one node to any other node, but no cycles or closed loops.
- **Minimum Spanning Tree:** A spanning tree with a minimal number of edges, which you can obtain by using a minimum distance criterion.
- Stepwise Optimal Hierarchical Clustering:

```
Begin: initialize $c, \hat{c} \leftarrow n, D_i \leftarrow \{\mathbf{x}_i\}, i = 1, \ldots, n$
    Do $\hat{c} \leftarrow \hat{c} - 1$
        Find clusters whose merger changes the criterion least, say $D_i$ and $D_j$
        Merge $D_i$ and $D_j$
    Until $c = \hat{c}$
Return $c$ clusters
```

- Clustering algorithms often use a confusion matrix measuring the dissimilarity of the data (induced metrics).

## Demonstrations



Misclassified samples: 81.0
Classification error: 20.2%

3. Stepping through iterations
Iteration 3

Mean for cluster 0: -0.48, 0.5

Covariance matrix:

```
0.03   0.0
0.0    0.03
```

Mean for cluster 1: 0.49, -0.51

Covariance matrix:

```
0.03   0.0
0.0    0.02
```

Overall results:
Total number of samples: 401.0
Misclassified samples: 1.0
Classification error: 0.25%

## Summary

- Reviewed the Bayes classifier.
- Revisited unsupervised Bayesian learning using a recursive approach.
- Compared ML and Bayes estimates.
- Compared supervised vs. unsupervised learning.
- Discussed computational considerations.
- Discussed hybrid methods (decision-directed) approaches to learning.
- Reviewed similarity measures and criterion functions.
- Discussed iterative optimization techniques.
- Introduced hierarchical clustering methods (agglomerative vs. divisive).
- Introduce the concept of spanning trees.