MACHINE LEARNING: CLUSTERING, AND CLASSIFICATION

Steve Tjoa
kiemyang@gmail.com

June 2013

Review from Day 2

Review from Day 2

Review from Day 2

- How did the lab go?

Review from Day 2

- How did the lab go?
- Did you try other audio files – other instrument recognizers?

Review from Day 2

- How did the lab go?
- Did you try other audio files – other instrument recognizers?
- Cross Validation?

Supervised vs. Unsupervised

- Unsupervised – "clustering"
- Supervised – binary classifiers (2 classes)
- Multiclass is derived from binary

Clustering

- Unsupervised learning – find pockets of data to group together

- Statistical analysis techniques

Clustering

- K = # of clusters

- Choosing the number of clusters – note that choosing the "best" number of clusters according to minimizing total squared distance will always result in same # of clusters as data points.

Clustering

The basic goal of clustering is to divide the data into groups such that the points within a group are close to each other, but far from items in other groups.

Hard clustering – each point is assigned to one and only one cluster.

K-Means

The key points relating to k-means clustering are:

K-Means

The key points relating to k-means clustering are:

- k-means is an automatic procedure for clustering unlabelled data;

K-Means

The key points relating to k-means clustering are:

- k-means is an automatic procedure for clustering unlabelled data;
- it requires a pre-specified number of clusters;

K-Means

The key points relating to k-means clustering are:

- k-means is an automatic procedure for clustering unlabelled data;
- it requires a pre-specified number of clusters;
- Clustering algorithm chooses a set of clusters with the minimum within-cluster variance

K-Means

The key points relating to k-means clustering are:

- k-means is an automatic procedure for clustering unlabelled data;
- it requires a pre-specified number of clusters;
- Clustering algorithm chooses a set of clusters with the minimum within-cluster variance
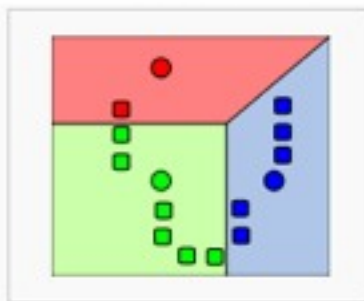- Guaranteed to converge (eventually)

Demo

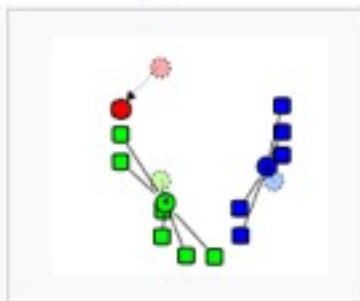- http://home.dei.polimi.it/matteucc/Clustering/tutorial_html/AppletKM.html

Demonstration of the standard algorithm
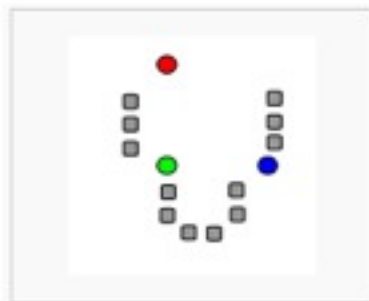
1) *k* initial "means" (in this case *k*=3) are randomly selected from the data set (shown in color).

2) *k* clusters are created by associating every observation with the nearest mean. The partitions here represent the Voronoi diagram generated by the means.

3) The centroid of each of the *k* clusters becomes the new means.

4) Steps 2 and 3 are repeated until convergence has been reached.

K-Means

The initialization method needs to be further specified. There are several possible ways to initialize the cluster centers:

K-Means

The initialization method needs to be further specified. There are several possible ways to initialize the cluster centers:

- Choose random data points as cluster centers

K-Means

The initialization method needs to be further specified. There are several possible ways to initialize the cluster centers:

- Choose random data points as cluster centers
- Randomly assign data points to K clusters and compute means as initial centers

K-Means

The initialization method needs to be further specified. There are several possible ways to initialize the cluster centers:

- Choose random data points as cluster centers
- Randomly assign data points to K clusters and compute means as initial centers
- Choose data points with extreme values

K-Means

The initialization method needs to be further specified. There are several possible ways to initialize the cluster centers:

- Choose random data points as cluster centers
- Randomly assign data points to K clusters and compute means as initial centers
- Choose data points with extreme values
- Find the mean for the whole data set then perturb into k means

ANALYSIS AND DECISION MAKING: GMMS

Mixture Models (GMM)

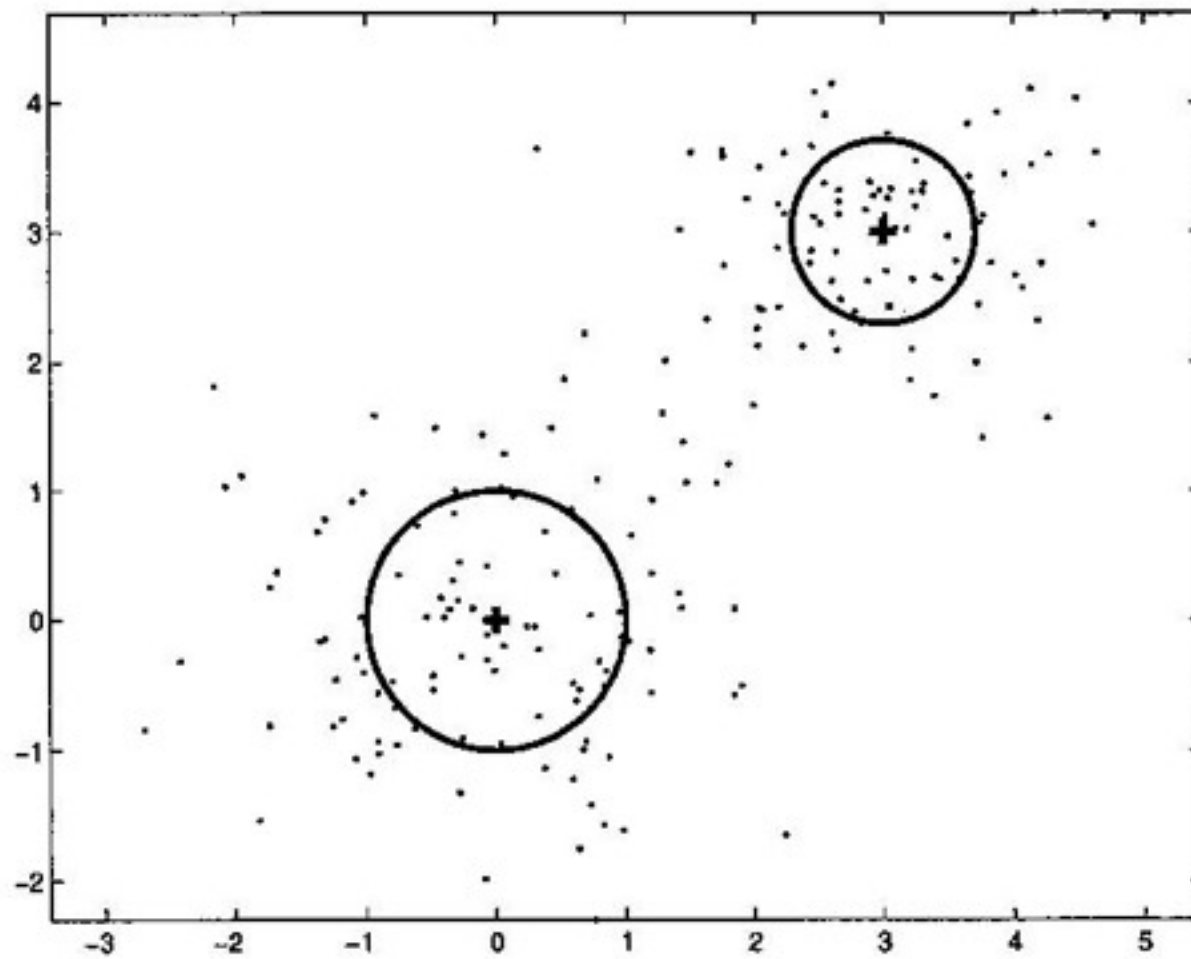- K-means = hard clusters.
- GMM = soft clusters.

Fig. 3.1. Spherical covariance mixture model. Sampled data (*dots*), **centres** (*crosses*) and one standard deviation error bars (*lines*).

Mixture Models (GMM)

- GMM is good because:
  1. Can approximate any pdf with enough components
  2. EM makes it easy to find components parameters
     - EM – the means and variances adapt to fit the data as well as possible
  3. Compresses data considerably

- Can make softer decisions (decide further downstream given additional information)

GMM Parameters

GMM Parameters

Input

GMM Parameters

Input
- Number of components (Gaussians)
  - e.g., 3

GMM Parameters

Input

- Number of components (Gaussians)
  - e.g., 3
- Mixture coefficients (sum = 1)
  - e.g., [0.5 0.2 0.3]
  - "Priors" or "Prior probabilities"
  - Priors are "the **original** probability that each point came from a given mixture."
  - "A prior is often the purely subjective assessment of an experienced expert."

GMM Parameters

Input
- Number of components (Gaussians)
  - e.g., 3
- Mixture coefficients (sum = 1)
  - e.g., [0.5 0.2 0.3]
  - "Priors" or "Prior probabilities"
  - Priors are "the **original** probability that each point came from a given mixture."
  - "A prior is often the purely subjective assessment of an experienced expert."
- Initialized centers, means, variances. (optional)

GMM Parameters

Input
- Number of components (Gaussians)
  - e.g., 3
- Mixture coefficients (sum = 1)
  - e.g., [0.5 0.2 0.3]
  - "Priors" or "Prior probabilities"
  - Priors are "the **original** probability that each point came from a given mixture."
  - "A prior is often the purely subjective assessment of an experienced expert."
- Initialized centers, means, variances. (optional)

Output

GMM Parameters

Input
- Number of components (Gaussians)
  - e.g., 3
- Mixture coefficients (sum = 1)
  - e.g., [0.5 0.2 0.3]
  - "Priors" or "Prior probabilities"
  - Priors are "the **original** probability that each point came from a given mixture."
  - "A prior is often the purely subjective assessment of an experienced expert."
- Initialized centers, means, variances. (optional)

Output
- Component centers/means, variances, and mixture coeff.

GMM Parameters

Input
- Number of components (Gaussians)
  - e.g., 3
- Mixture coefficients (sum = 1)
  - e.g., [0.5 0.2 0.3]
  - "Priors" or "Prior probabilities"
  - Priors are "the **original** probability that each point came from a given mixture."
  - "A prior is often the purely subjective assessment of an experienced expert."
- Initialized centers, means, variances. (optional)

Output
- Component centers/means, variances, and mixture coeff.
- Posterior probabilities
  - "Posterior probabilities are the responsibilities which the Gaussian components have for each of the data points."

GMM Parameters

Input
- Number of components (Gaussians)
  - e.g., 3
- Mixture coefficients (sum = 1)
  - e.g., [0.5 0.2 0.3]
  - "Priors" or "Prior probabilities"
  - Priors are "the **original** probability that each point came from a given mixture."
  - "A prior is often the purely subjective assessment of an experienced expert."
- Initialized centers, means, variances. (optional)

Output
- Component centers/means, variances, and mixture coeff.
- Posterior probabilities
  - "Posterior probabilities are the responsibilities which the Gaussian components have for each of the data points."

Query

GMM Parameters

Input
- Number of components (Gaussians)
  - e.g., 3
- Mixture coefficients (sum = 1)
  - e.g., [0.5 0.2 0.3]
  - "Priors" or "Prior probabilities"
  - Priors are "the **original** probability that each point came from a given mixture."
  - "A prior is often the purely subjective assessment of an experienced expert."
- Initialized centers, means, variances. (optional)

Output
- Component centers/means, variances, and mixture coeff.
- Posterior probabilities
  - "Posterior probabilities are the responsibilities which the Gaussian components have for each of the data points."
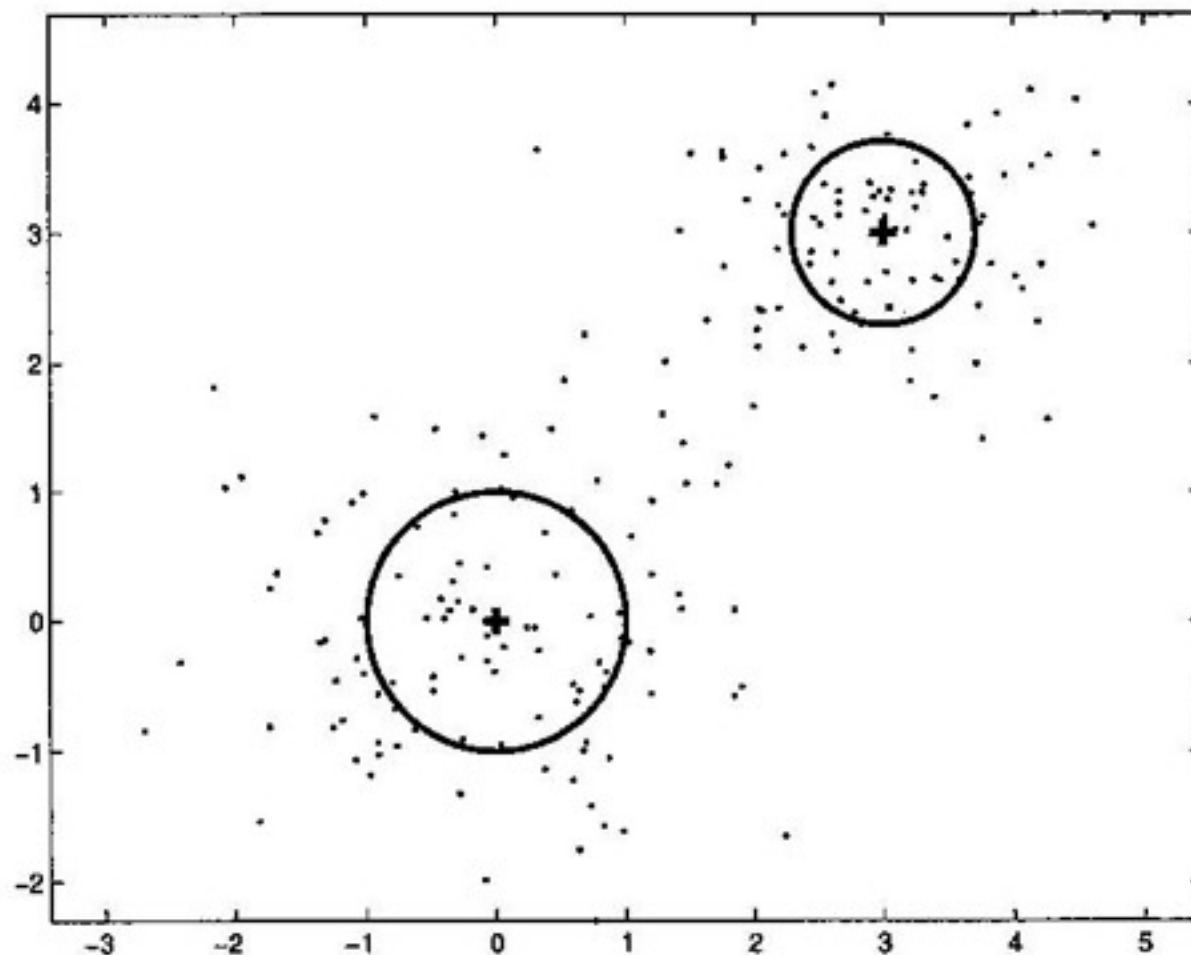
Query
- Obtain similarity via Likelihood

Fig. 3.1. Spherical covariance mixture model. Sampled data (*dots*), **centres** (*crosses*) and one standard deviation error bars (*lines*).
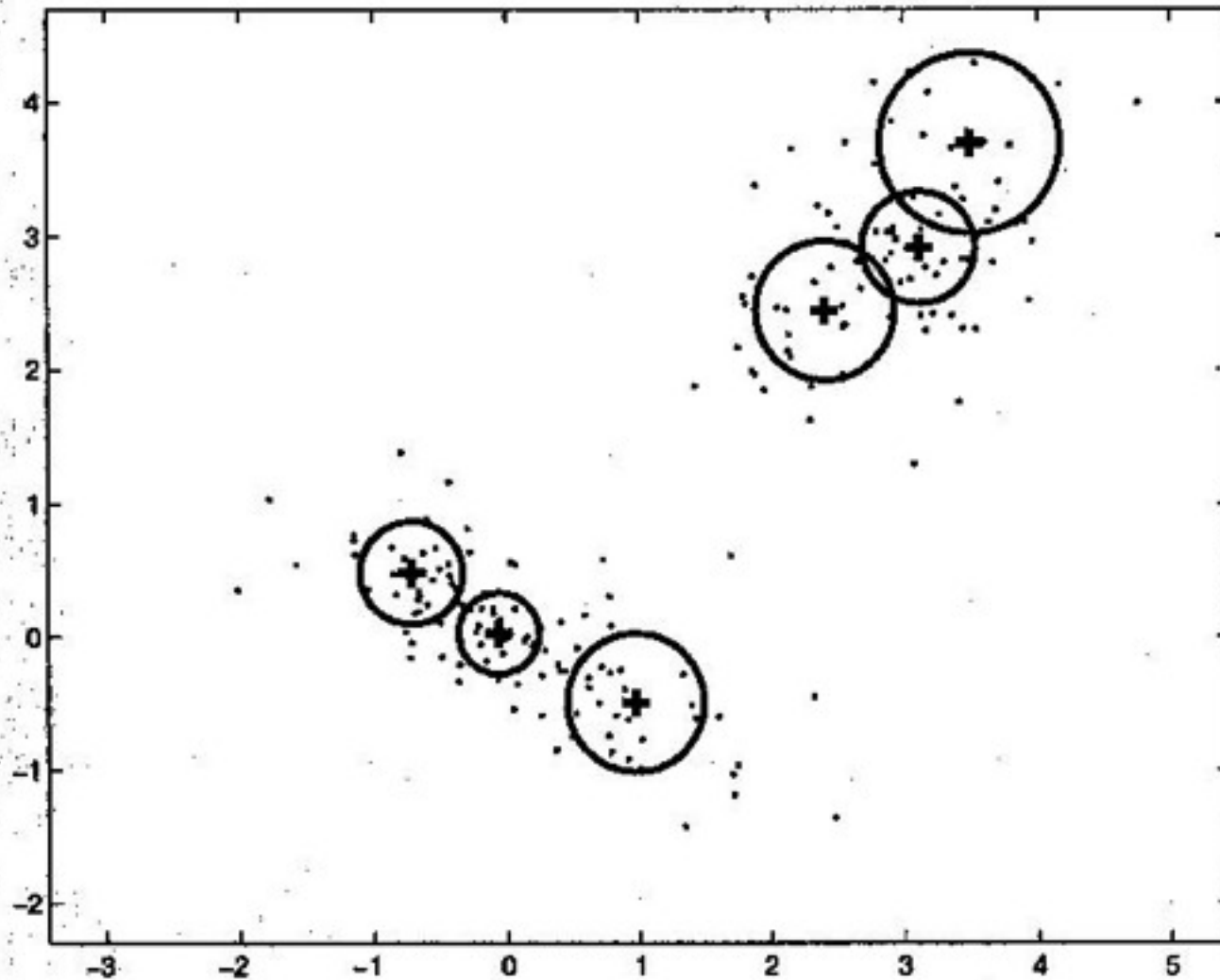
4. Spherical covariance mixture model with six components fitted to the mpled from the full covariance two-component model in Fig. 3.3. Sampled ots), centres (*crosses*) and one standard deviation error bars (*lines*).
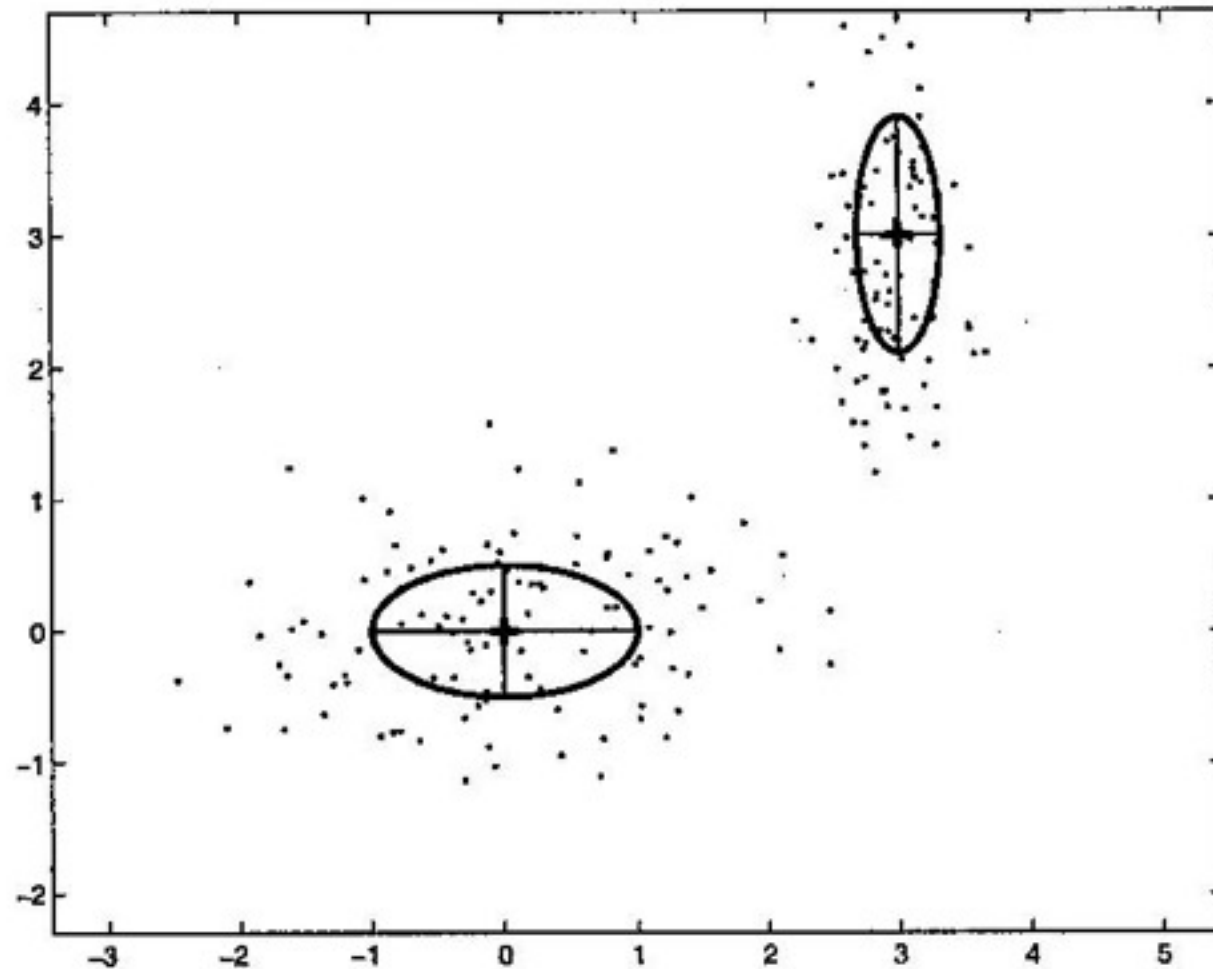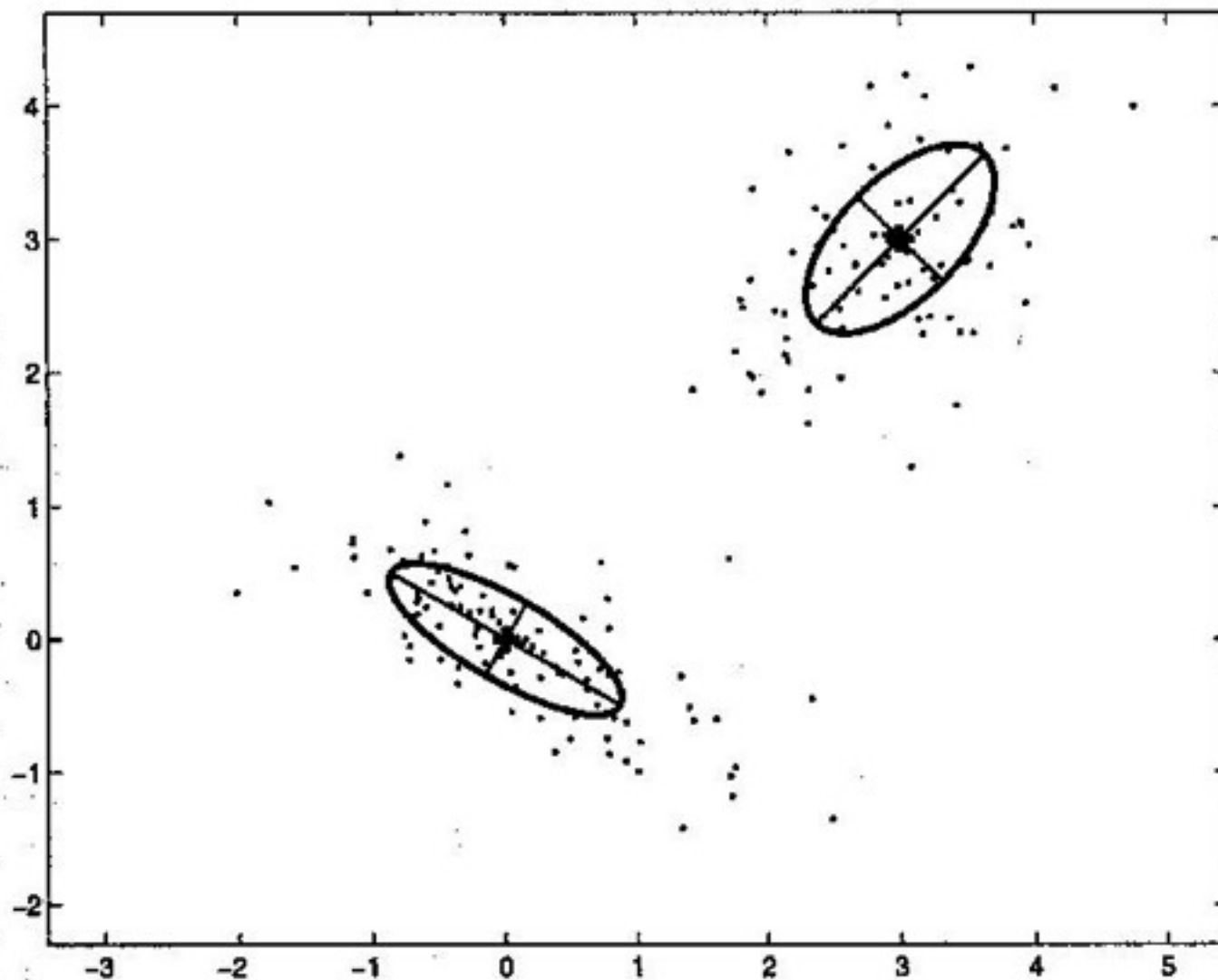
- From Netlab (p82–83)

Fig. 3.2. Diagonal covariance mixture model. Sampled data (*dots*), centres (*crosses*), covariance axes (*thin lines*) and one standard deviation error bars (*thick lines*).

3. Full covariance mixture model. Sampled data (*dots*), centres (*crosses*), nce axes (*thin lines*) and one standard deviation error bars (*thick lines*).

GMM

- "Pooled covariance" – using a single covariance to describe all clusters (saves on parameter computation)

GMM: Likelihood

1. Evaluate the probability of that mixture
   modeling your point.

GMM: Likelihood

1. Evaluate the probability of that mixture modeling your point.
   likelihoodgm1 = gmmprob (gm1,testing_features)

GMM: Likelihood

1. Evaluate the probability of that mixture modeling your point.

    likelihoodgm1 = gmmprob (gm1,testing_features)
    likelihoodgm2 = gmmprob (gm2,testing_features);

GMM: Likelihood

1. Evaluate the probability of that mixture modeling your point.
   likelihoodgm1 = gmmprob (gm1,testing_features)
   likelihoodgm2 = gmmprob (gm2,testing_features);
   loglikelihood = log(likelihoodKick ./ likelihoodSnare)

GMM: Likelihood

1. Evaluate the probability of that mixture modeling your point.

    likelihoodgm1 = gmmprob (gm1,testing_features)

    likelihoodgm2 = gmmprob (gm2,testing_features);

    loglikelihood = log(likelihoodKick ./ likelihoodSnare)

GMM: Likelihood

1. Evaluate the probability of that mixture modeling your point.

    likelihoodgm1 = gmmprob (gm1,testing_features)
    likelihoodgm2 = gmmprob (gm2,testing_features);
    loglikelihood = log(likelihoodKick ./ likelihoodSnare)

- Log-function is "order-preserving" – maximizing a function vs. maximizing its log

Minimization Problems

>Demgmm1

Minimization Problems

>Demgmm1

- EM is gradient-based – it does not find the global maximum in the general case, unless properly initialized in the general region of interest.

Minimization Problems

>Demgmm1

- EM is gradient-based – it does not find the global maximum in the general case, unless properly initialized in the general region of interest.

Minimization Problems

>Demgmm1

- EM is gradient-based – it does not find the global maximum in the general case, unless properly initialized in the general region of interest.

- Error wants to be –inf, which occurs when Gaussian is fit for each data point. (mean = data point and variance = 0)

Minimization Problems

>Demgmm1

- EM is gradient-based – it does not find the global maximum in the general case, unless properly initialized in the general region of interest.

- Error wants to be –inf, which occurs when Gaussian is fit for each data point. (mean = data point and variance = 0)

Minimization Problems

>Demgmm1

- EM is gradient-based – it does not find the global maximum in the general case, unless properly initialized in the general region of interest.

- Error wants to be –inf, which occurs when Gaussian is fit for each data point. (mean = data point and variance = 0)

- "There are often a large number of local minima which correspond to poor models. Solution is to build models from many different initialization

GMM

- Application:
  - State-of-the-art speech recognition systems
  - estimate up to 30,000 separate GMMs, each with about 32 components. This means that these systems can have up to a million Gaussian components!! All the parameters are estimated from (a lot of) data by the EM algorithm.

  Application:
  Speaker Recognition

GMM System

- **Separate models for** $p(x|sing)$, $p(x|no\ sing)$
  - combined via likelihood ratio test
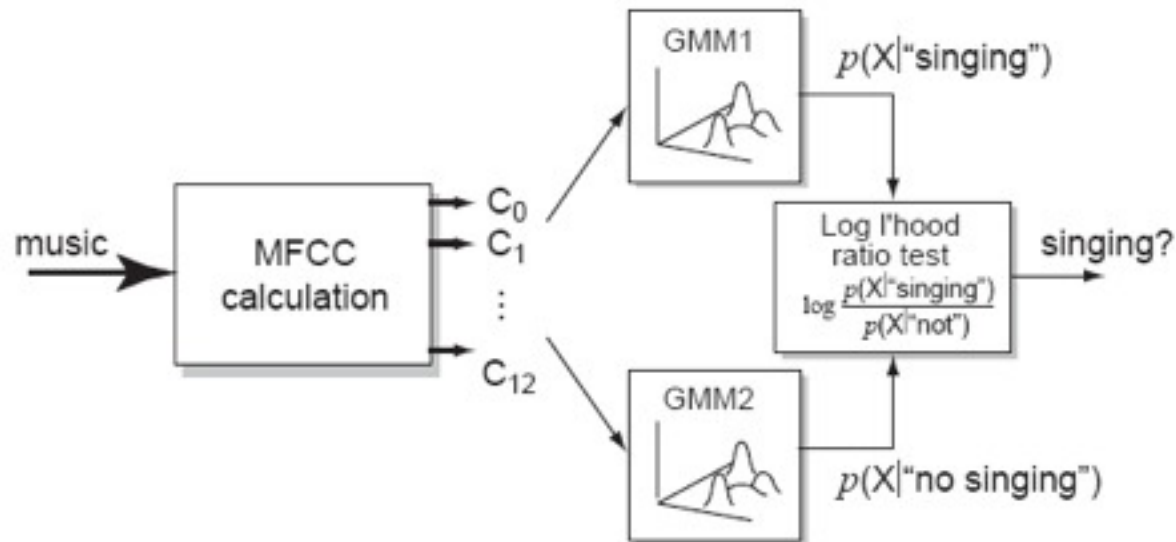
- **How many Gaussians for each?**
  - say 20; depends on data & complexity

- **What kind of covariance?**
  - diagonal (spherical?)

Thursday, June 27, 2013

Genre

"Because feature vectors are computed from short segments of audio, an entire song induces a cloud of points in feature space."

Genre

"Because feature vectors are computed from short segments of audio, an entire song induces a cloud of points in feature space."

"The cloud can be thought of as samples from a distribution that characterizes the song, and we can model that distribution using statistical techniques. Extending this idea, we can conceive of a distribution in feature space that characterizes the entire repertoire of each artist."

Genre

"Because feature vectors are computed from short segments of audio, an entire song induces a cloud of points in feature space."

"The cloud can be thought of as samples from a distribution that characterizes the song, and we can model that distribution using statistical techniques. Extending this idea, we can conceive of a distribution in feature space that characterizes the entire repertoire of each artist."

Genre

"Because feature vectors are computed from short segments of audio, an entire song induces a cloud of points in feature space."

"The cloud can be thought of as samples from a distribution that characterizes the song, and we can model that distribution using statistical techniques. Extending this idea, we can conceive of a distribution in feature space that characterizes the entire repertoire of each artist."

A. Berenzweig, B. Logan, D. Ellis, and B. Whitman. A large-scale evalutation

Genre

"Because feature vectors are computed from short segments of audio, an entire song induces a cloud of points in feature space."

"The cloud can be thought of as samples from a distribution that characterizes the song, and we can model that distribution using statistical techniques. Extending this idea, we can conceive of a distribution in feature space that characterizes the entire repertoire of each artist."
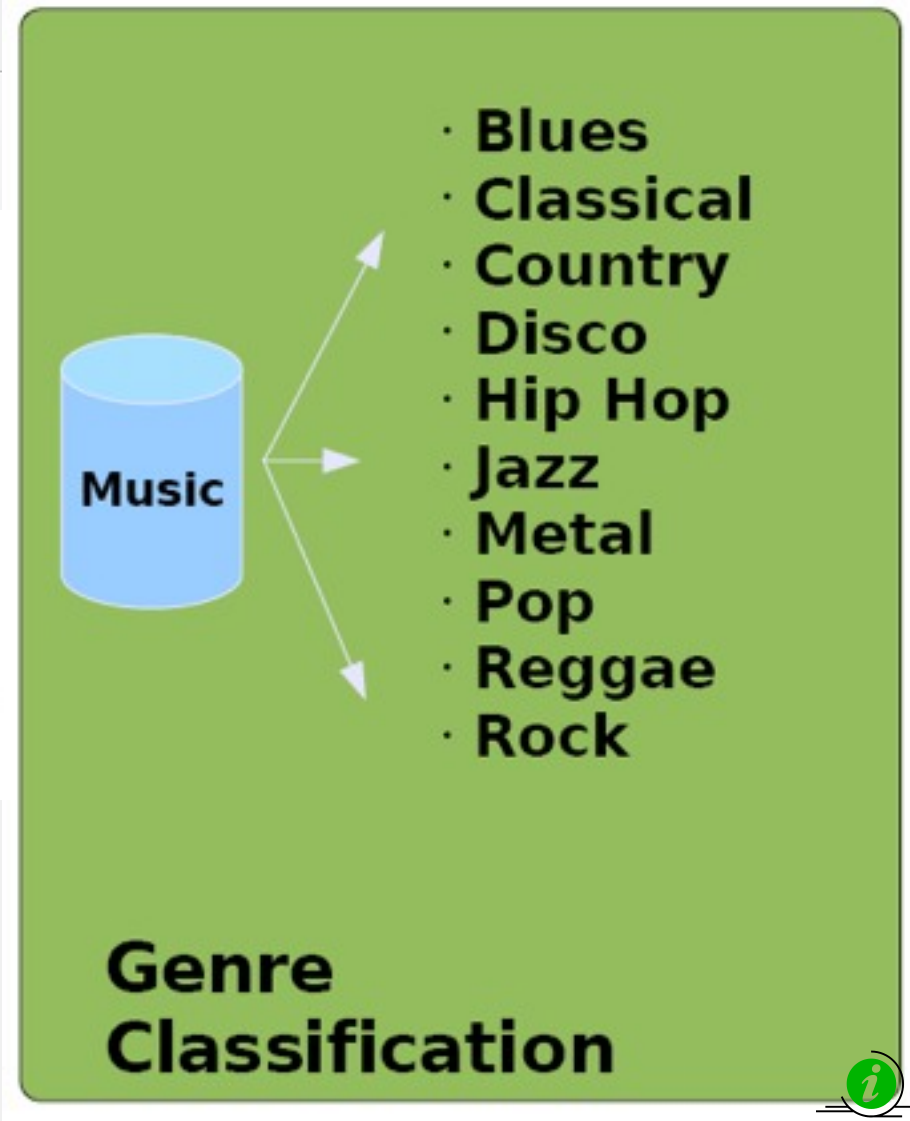
A. Berenzweig, B. Logan, D. Ellis, and B. Whitman. A large-scale evalutation of acoustic and subjective music similarity measures. In Proceedings

Genre

"Because feature vectors are computed from short segments of audio, an entire song induces a cloud of points in feature space."

"The cloud can be thought of as samples from a distribution that characterizes the song, and we can model that distribution using statistical techniques. Extending this idea, we can conceive of a distribution in feature space that characterizes the entire repertoire of each artist."

A. Berenzweig, B. Logan, D. Ellis, and B. Whitman. A large-scale evalutation of acoustic and subjective music similarity measures. In Proceedings of 4th International Symposium on Music Information Retrieval,

- **Genre Classification**:
  - Manual :   72%
    (Perrot/Gjerdigen)
  - Automated (2002)   60%
    (Tzanetakis)
  - Automated (2005)   82%
    (Bergstra/Casagrande/Eck)
  - Automated (2007)   76%

*From ISMIR 2007 Music Recommender
Tutorial (Lamere & Celma)*

Music

· **Blues**
· **Classical**
· **Country**
· **Disco**
· **Hip Hop**
· **Jazz**
· **Metal**
· **Pop**
· **Reggae**
· **Rock**

**Genre
Classification**

Thursday, June 27, 2013

How?

How?

- Version 1 – One feature vector per song
  - High-level features extracted from data
    - Timbral (MFCCs, etc), Rhythmic content (beat histogram, autocor, tempos), Pitch info
    - Sampling of the frames in the song
  - Statistics of features extracted from a piece (includes means, weights, etc)
  - Representative of MFCC spectral shape
  - Could further use "Anchor space" where classifiers are training to represent musically meaningful classifiers. (Euclidean distance between anchor space)

How?

- Version 1 – One feature vector per song
  - High-level features extracted from data
    - Timbral (MFCCs, etc), Rhythmic content (beat histogram, autocor, tempos), Pitch info
    - Sampling of the frames in the song
  - Statistics of features extracted from a piece (includes means, weights, etc)
  - Representative of MFCC spectral shape
  - Could further use "Anchor space" where classifiers are training to represent musically meaningful classifiers. (Euclidean distance between anchor space)

- Version 2 – Cloud of points
  - Extract audio every N frames
  - K-Means or GMM representing a "cloud of points" for song
    - Clusters: mean, covariance and weight of each cluster = signature for song/artist/genre

Music Recommendation

- Cloud of points from frames of song
  - High-level features extracted from data
  - Classifier: Weighted attribute nearest neighbors or fast distance measures.
  - k-Means or GMM used to create clusters.
  - The mean, covariance and weight of each cluster = signature for the song.

  - Compare distance between other songs (signature) using various techniques to measure distance between probability distributions. (Most similar = closest distance)

Automatic annotation
Similarity based on classification
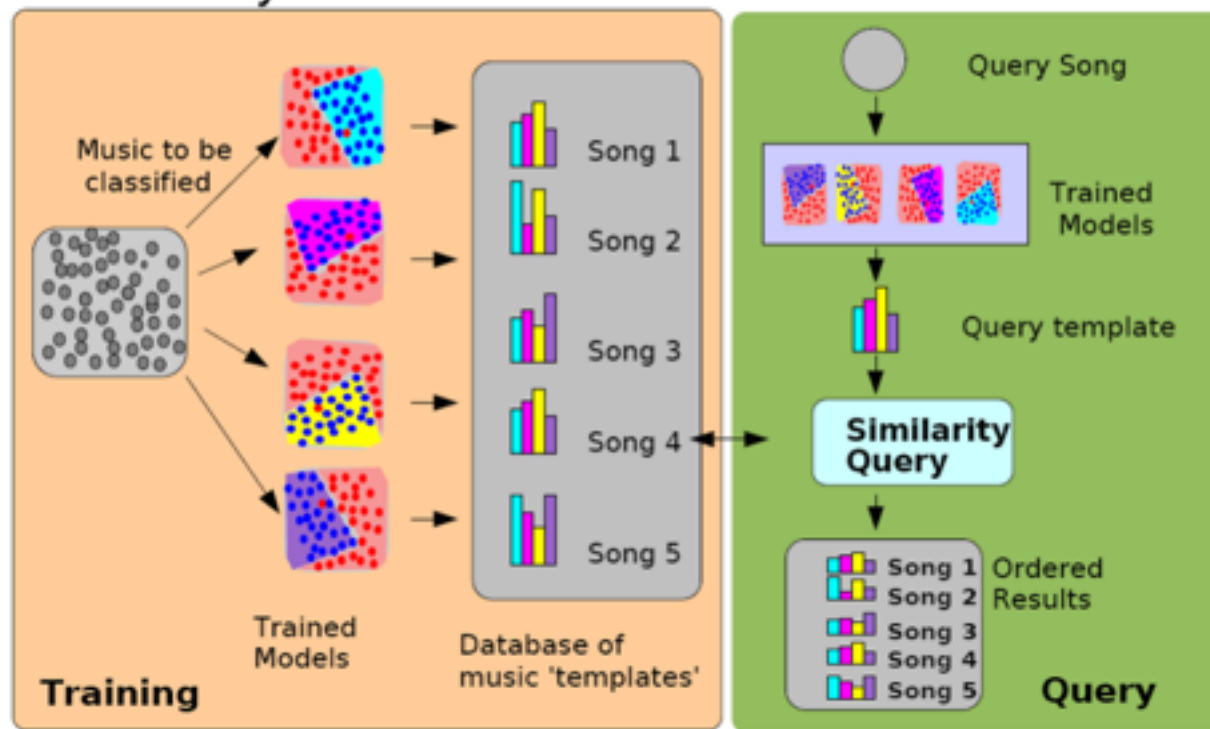
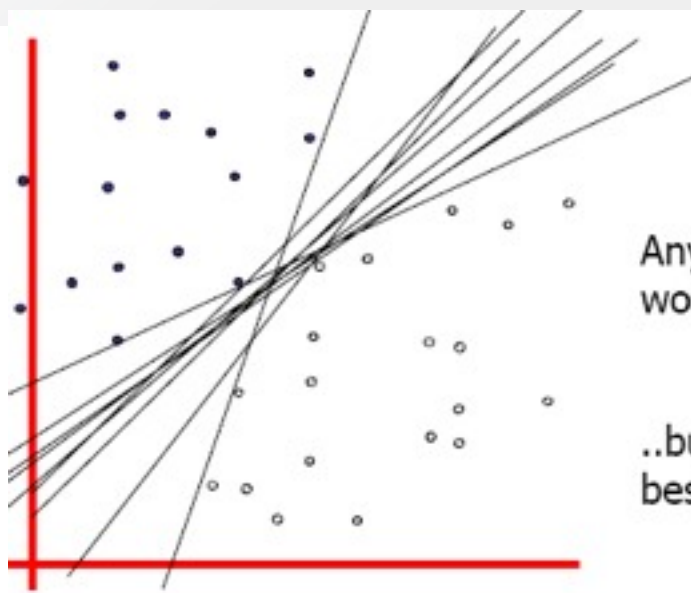*From ISMIR 2007 Music Recommender Tutorial (Lamere & Celma)*
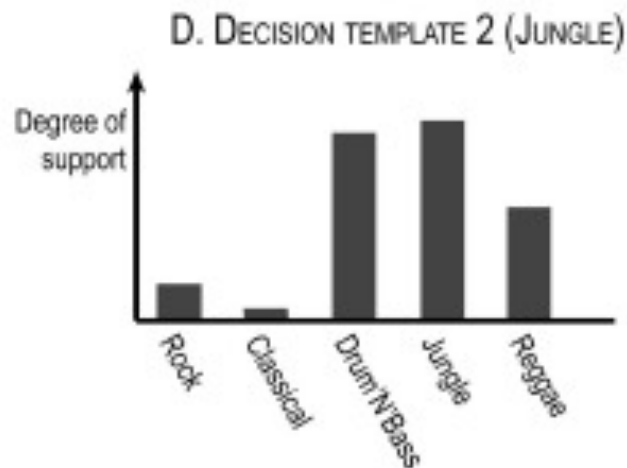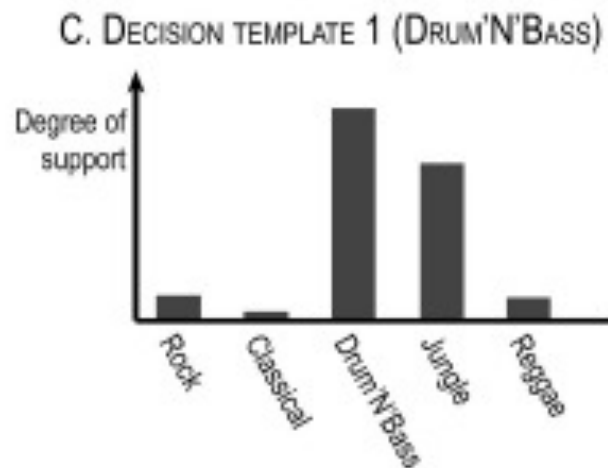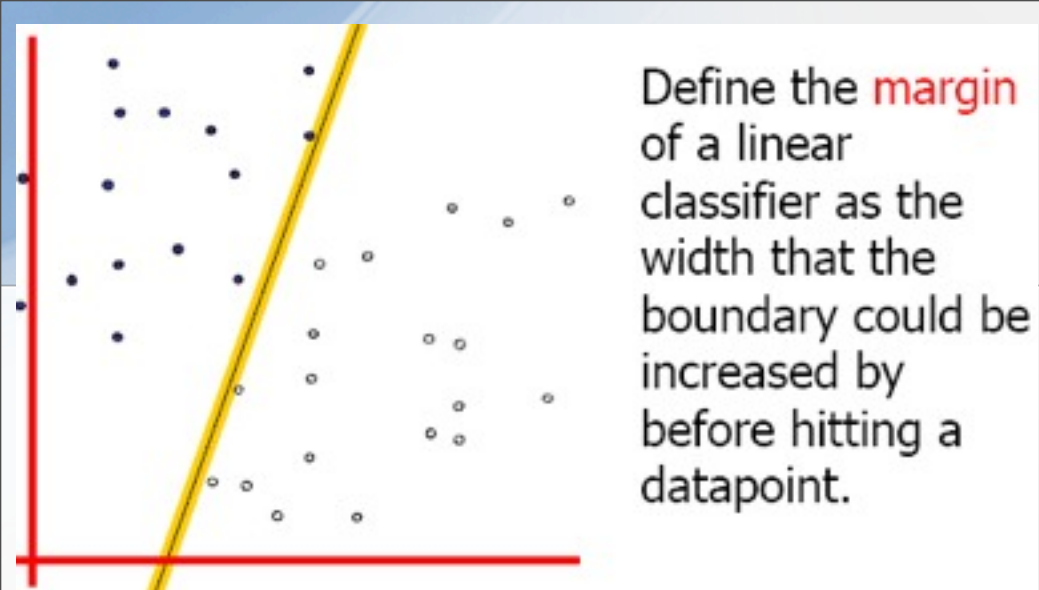
SUPPORT VECTOR MACHINES (SVM)
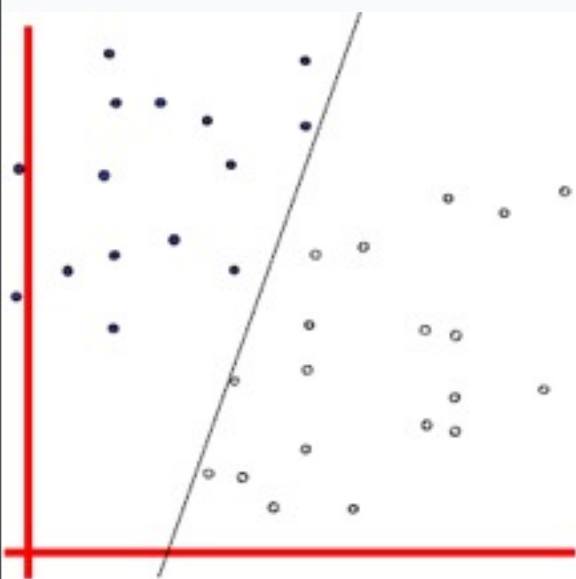
How would you classify this data?

Any of these would be fine..
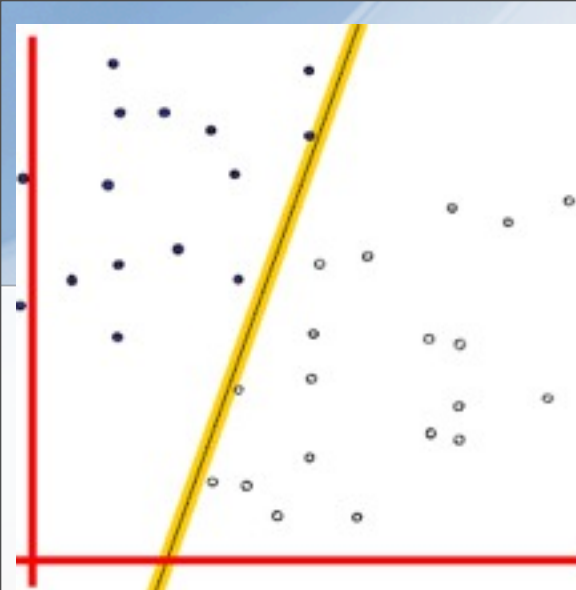
..but which is best?

From : http://www.autonlab.org/tutorials/

Define the margin of a linear classifier as the width that the boundary could be increased by before hitting a datapoint.
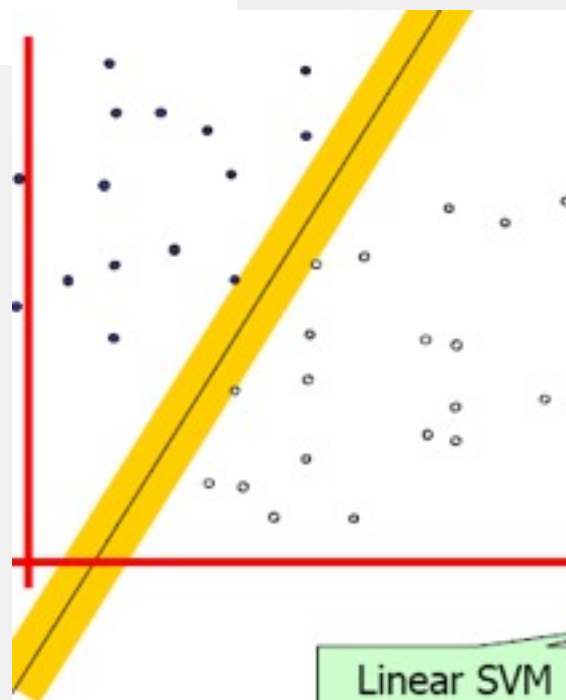
From : http://www.autonlab.org/tutorials/

Define the **margin** of a linear classifier as the width that the boundary could be increased by before hitting a datapoint.

The **maximum margin linear classifier** is the linear classifier with the, um, maximum margin.

This is the simplest kind of SVM (Called an LSVM)

Linear SVM

From : http://www.autonlab.org/tutorials/

SVM

- Hyperplane separates the data from the two classes with a "maximum margin".
- Support Vectors – are those data points that the margin pushes up against
- SVM training is guaranteed to find the global minimum of the cost function.
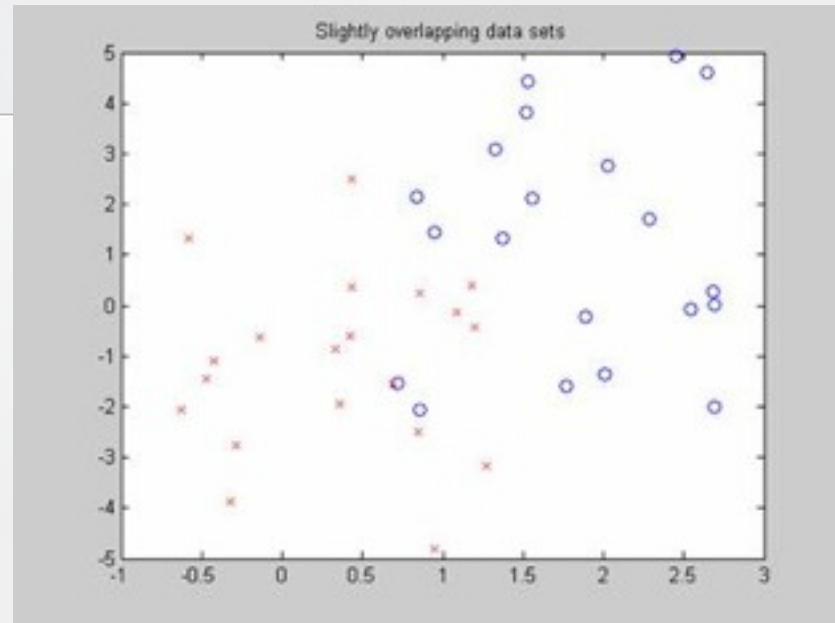- Less experience needed – fewer parameters to tune

SVM with polynomial kernel visualization by Udi Aharoni

- http://www.youtube.com/watch?v=3liCbRZPrZA

feature 2

feature 1

SVM Parameters
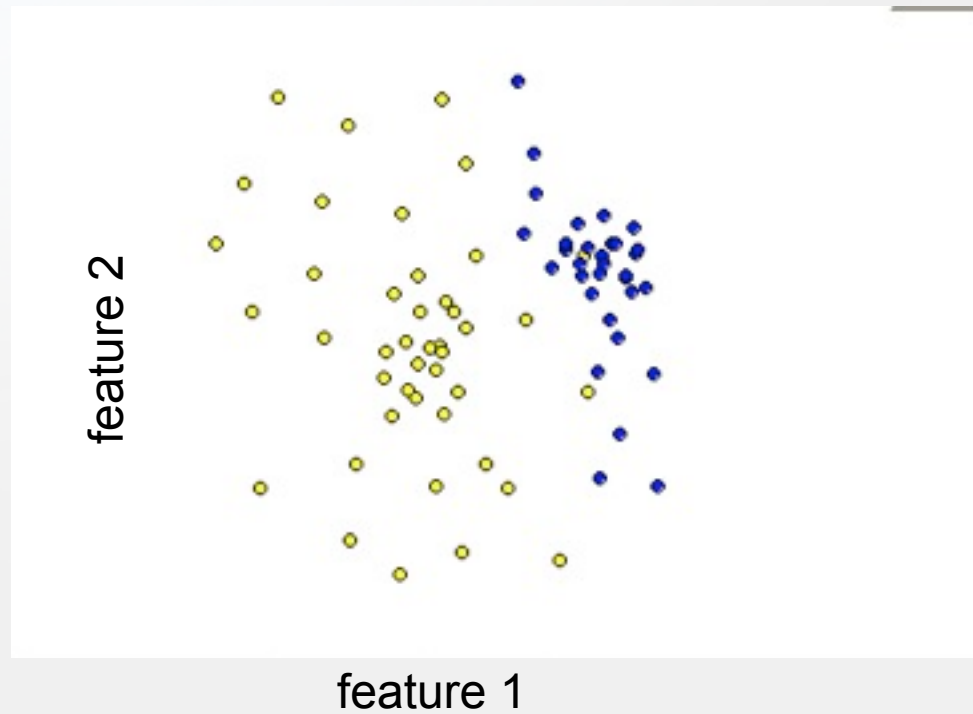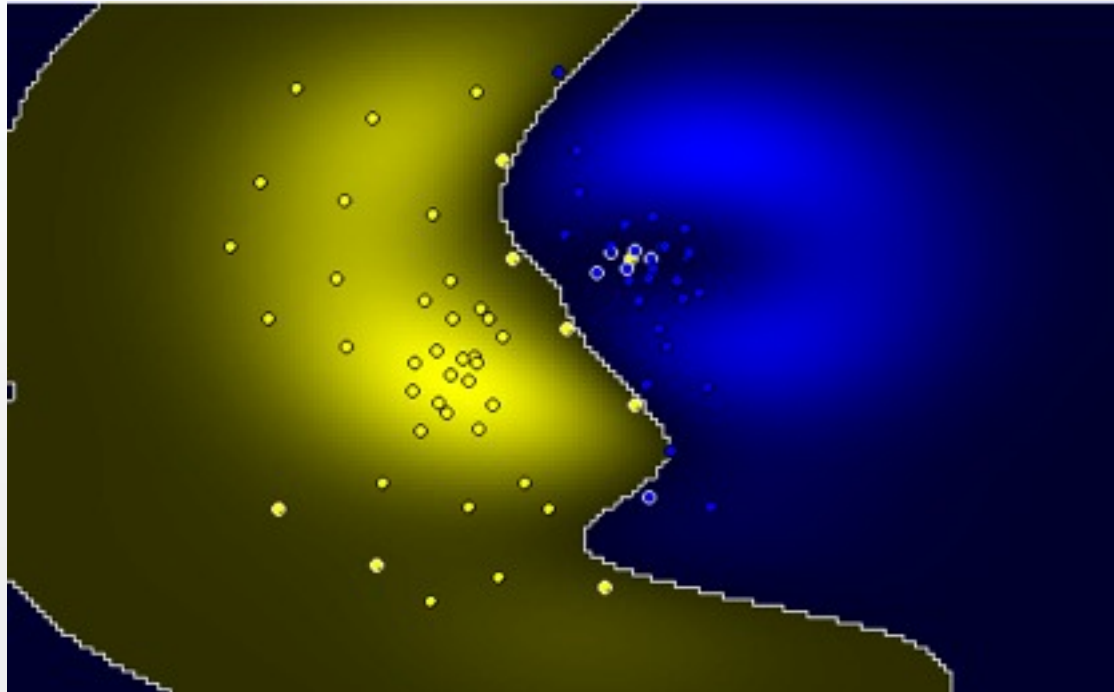
Slightly overlapping data sets

What effect do the parameters of an radial–basis–function SVM have on the separating the two data sets?

Using the RBF kernel, we have to choose values of :

gamma   = degree of curviness of the hyperplane / complexity of the contour

C   = allowance for points to overlap into each other's class

Video 1
Video 2

RBF Parameters: C and gamma

- Grid search using cross-validation to find the best one. Coarse then fine grid search.
- e.g., 2-5, 2-3, ... 2+15, gamma = 2-15, 2-13, 2+3

RBF Parameters: C and gamma

- Grid search using cross-validation to find the best one. Coarse then fine grid search.
- e.g., 2-5, 2-3, ... 2+15, gamma = 2-15, 2-13, 2+3
- Why grid search
  - Psychological (If you have time for brute force... why chance it on approximations or heuristics)
  - Since there are only 2 params, grid search isn't all the different from advanced estimation techniques
  - Easily parallelized (C and gamma are

Practical Guide to SVM: The Lab

- Feature selection?
- Scale feature data
  - Save scaling stats so we can scale the test data to be in the same range
- Feature format
- Class labels {1,-1} or {0,1}
- Kernels (linear, polynomial, RBF, sigmoid)
- Find best C and gamma (cross-validation)
- Train with entire training set
- Test with validation or test set

- easy.py or grid.py