The Automatic Musicologist

Douglas Turnbull

Department of Computer Science and Engineering

University of California, San Diego

UCSD AI Seminar

April 12, 2004

Based on the paper:

"Fast Recognition of Musical Genre using RBF Networks"

By Douglas Turnbull & Charles Elkan

Human Music Classification:

For us, this is a pretty simple task.

Let's see some examples:

1. The King and the Thief
2. Amazon Jungle

Maybe music classification is not as simple as it seems.

- We don't always agree on the genre of a song.
- We don't even agree on the set of genres.

These two issues are debated by musicologists, cognitive scientist, and music fans alike.

Human Music Classification:

Classification of music by genre is difficult to automate due to the subjective nature of music.

Our perception of sound is influenced by **memory**, **emotions**, and **social context**.

Creating a deep representation of **emotion** or **social context** is beyond the reach of current AI methods

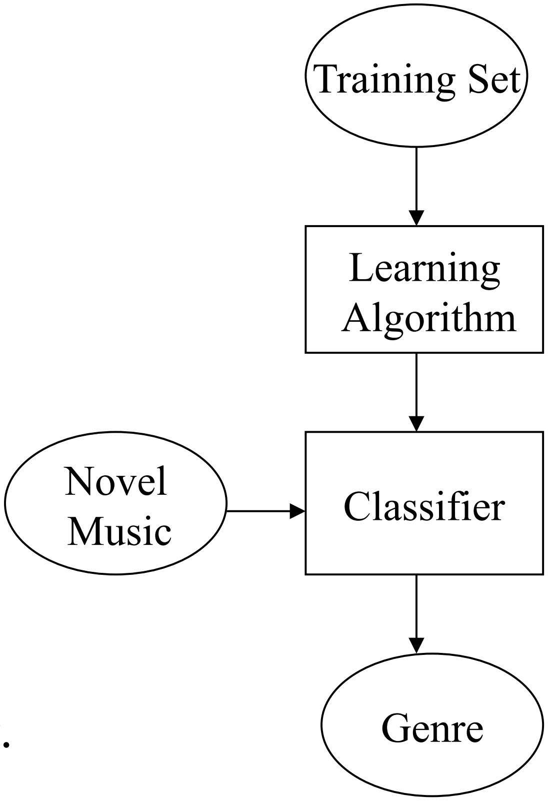
But maybe we can mimic **auditory memory**.

Automatic Music Classification:

Goal: extract information from previously heard audio tracks in order to recognize the genre of new tracks.

This is an example "The Learning Problem" described in slide 1, lecture 1 of Prof. Dasgupta class on Machine Learning (CSE250B, Spring 2004)

Input space:	Audio Tracks
Output space:	Musical Genre
Training Set:	Human Labeled Audio Tracks
Classifier:	Radial Basis (RBF) Networks



We use novel audio samples to evaluate our classifier.

Audio Feature Extraction:

CD quality audio has 44,100 16-bit samples per second. Our 30-second feature vector would be:

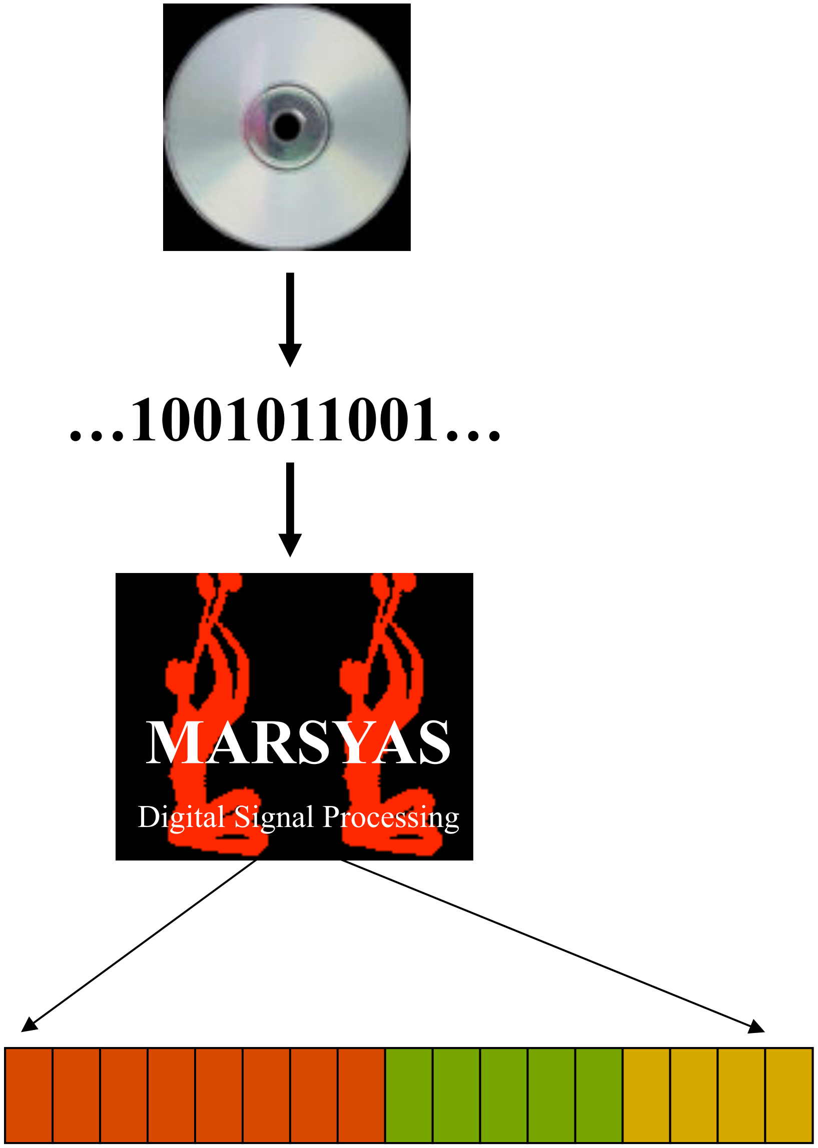
$$X = \{0,\ldots,65535)^{30*44,100}$$

Our first task is to reduce the dimensionality using digital signal processing.

We will use the MARSYAS software to extract 30 real value measurements from each audio track.

$$X = \{\mathrm{Real}\}^{30}$$

Audio Feature Extraction:



music:

digital signal: **…1001011001…**

feature extraction:

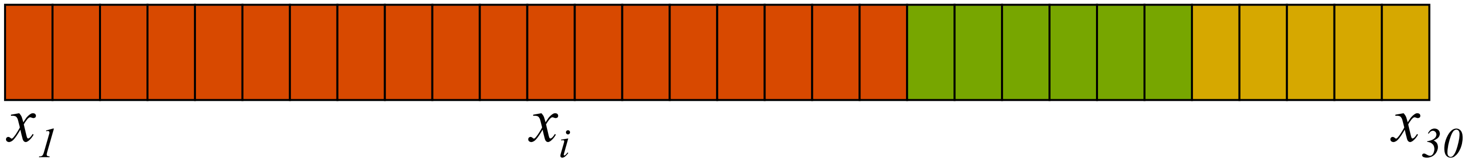
feature vector:

MARSYAS

Extraction of 30 features from 30-second audio tracks

Timbral Texture (19)

- Music-Speech discrimination, Speech Recognition
  - Short Time Fourier Transform (STFT) algorithm
- Examples – means and variances
  - Spectral Centroid – ‘brightness’ of sound
  - Spectral Flux – local spectral change
  - Zero Crossings – ‘noisiness’ of signal
  - Low-Energy – amount of quiet time
  - Mel-frequency Cepstral Cooefficients (MFCC)



MARSYAS

Extraction of 30 features from 30-second audio tracks

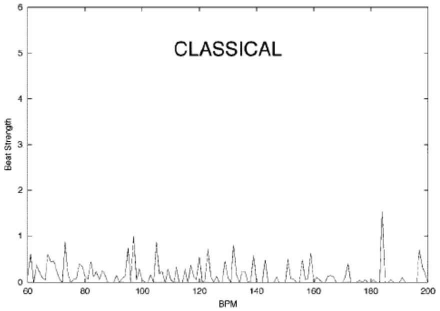
Timbral Texture (19)

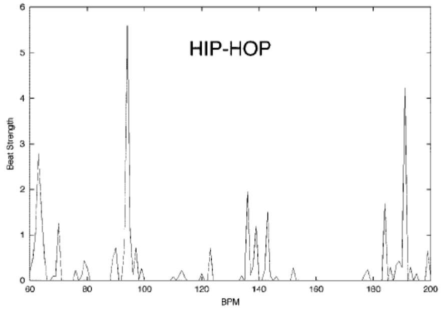
Rhythmic Content (6)

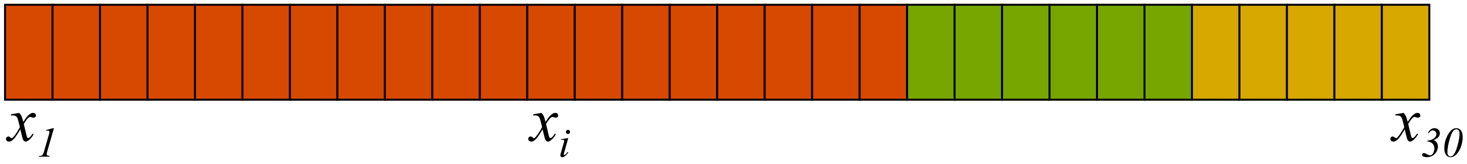
- Beat Strength, Amplitude, Tempo Analysis
  - •Wavelet Tansform

•Examples

- Frequencies of peaks
- Relative amplitude of major peaks
- Sum of all peaks







MARSYAS

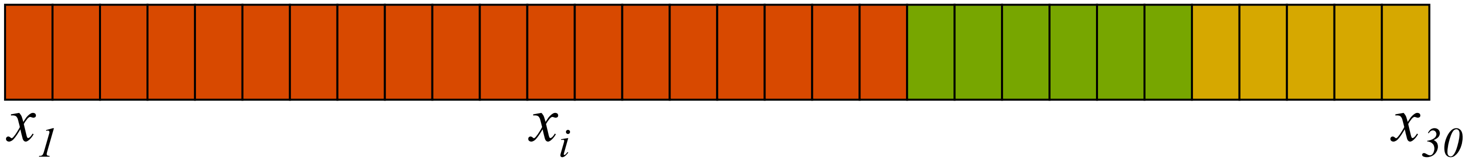
Extraction of 30 features from 30-second audio tracks

Timbral Texture (19)

Rhythmic Content (6)

Pitch Content (5)

- Dominant Pitch , Pitch Intervals
  - Multipitch Detection Algorithm
- Examples
  - Frequency of highest peak
  - Amplitude of highest peak
    - **Large** for tonal music (ex. Rock and HipHop)
  - Intervals between peaks



The Data Set:

The data set created by Tzanetakis and Cook[1] uses 10 genres, each of which have 100 examples.

The genres included are:

- Classical
- Country
- Disco
- Hip-Hop
- Jazz
- Rock
- Blues
- Reggae
- Pop
- Metal

The assigned class labels are mutually exclusive and have a uniform strength of assignment.

Classification:

Input space: Audio Tracks {30-Dimension feature vector}
Output space: Musical Genre {Classical, … , Metal}

Training Set: Human Labeled Audio Tracks
Classifier: **Radial Basis (RBF) Networks**

Before I can discuss RBF networks, we need to introduce the concept of Radial Basis Functions.

Radial Basis Functions (RBFs) :

An RBF measures how far an input vector (***x***) is from a prototype vector (**μ**). We use unnormalized, spherical Gaussians for our basis functions.

$$\phi_j(\boldsymbol{x}) = exp\{-\frac{||\boldsymbol{x} - \boldsymbol{\mu}_j||^2}{2\sigma_j^2}\}$$

We know that ***x*** is audio feature vector.

What are the vector ***μ*** and scalar *σ* parameter?

- They represent a ‘center’ and ‘spread’ of data points in some region of the input space
- They can be initialized using a number of methods
- They can be adjusted during training

Initializing Radial Basis Functions - Unsupervised:

1. K-means (KM)

We use a special form of K-means call "Subset Furthest-First K-means:"

- Randomly select subset of O(*k log k*) data point
- Find initial centers by taking data points from the subset that are far apart.
- Run K-mean algorithm on all data point

Upon convergence, each of the *k* cluster centers represents a prototype vector $\boldsymbol{\mu}$**.** We set σ to be the standard deviation of the distances from $\boldsymbol{\mu}$ to all other points assigned to that cluster.

In addition, we can make use of the triangle inequality to reduce the number of distance calculations in order to make the algorithm run faster.

Initializing Radial Basis Functions - Supervised:

2. Maximum Likelihood for Gaussians (MLG)

- For each class, the prototype vector ***μ*** is the average of the data points assigned to that class.

$$\mu_{Class_k} = \frac{1}{|Class_k|} \sum_{x^{(n)} \in Class_k} x^{(n)}$$

- We set *σ* to be the standard deviation of the distances from ***μ*** to all other points in the class.

$$\sigma^2_{Class_k} = \frac{1}{|Class_k|} \sum_{x^{(n)} \in Class_k} ||x^{(n)} - \mu_k||^2$$

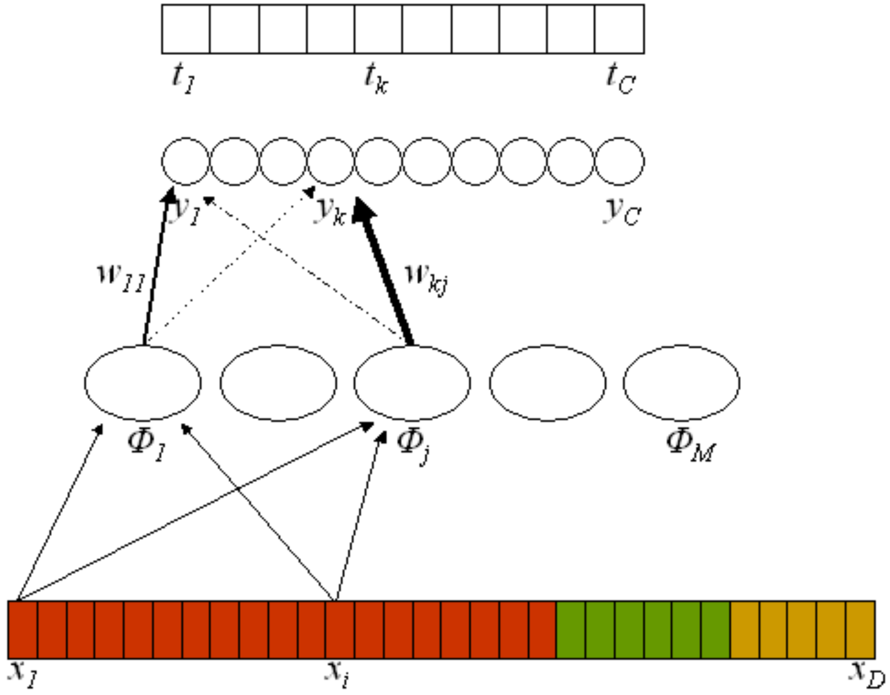
3. In-class K-means (ICKM)

- For each class c_k, K-means algorithm is run on the data points with class label c_k.

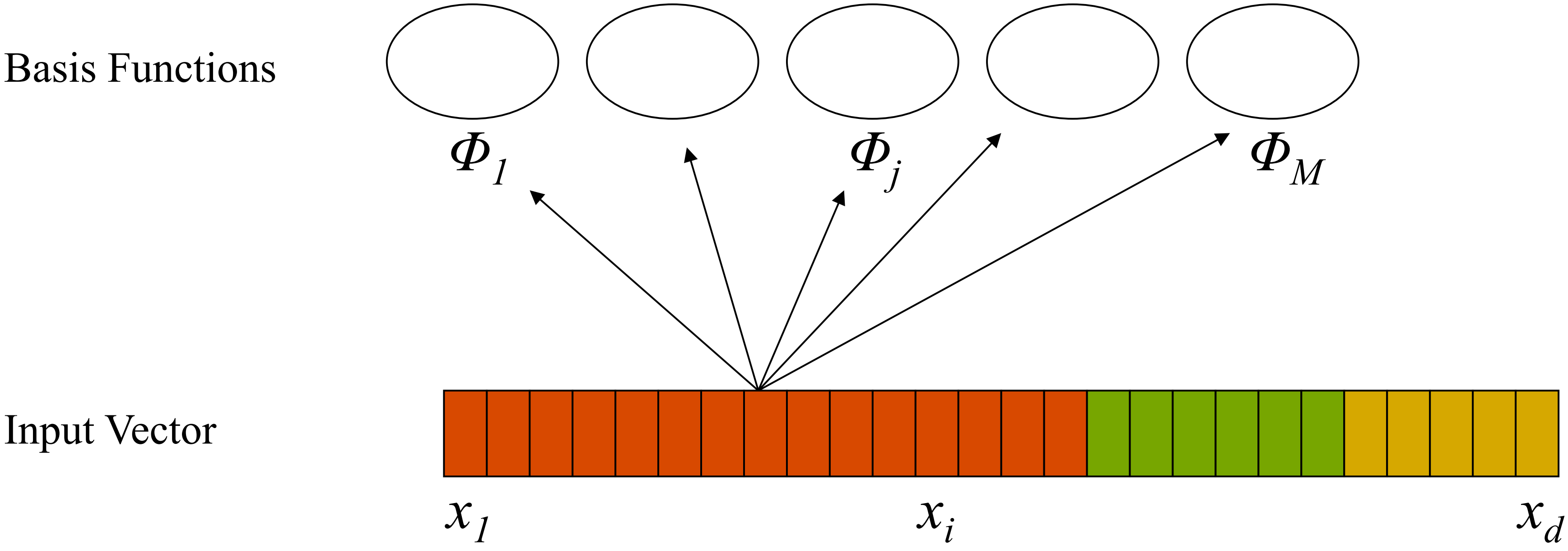
RBF Networks

A Radial Basis Function (RBF) network is a two-layer, feed-forward neural network. The two layers are:

- RBF layer
- Linear Discriminant layer



The RBF Layer



Basis Functions

Φ_1 Φ_j Φ_M

Input Vector

x_1 x_i x_d

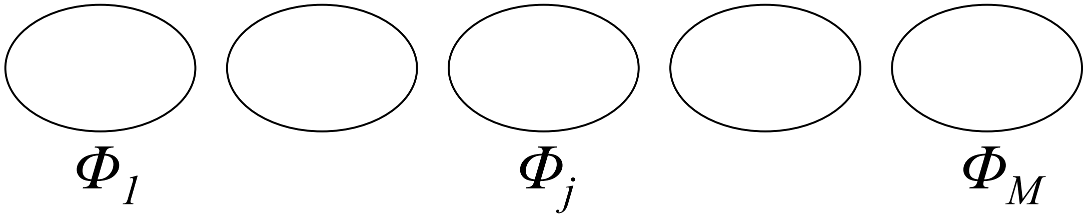
If the input vector $\boldsymbol{x}$ is close to the prototype vector $\boldsymbol{\mu_j}$ of the jth basis function Φ_j, then $\Phi_j(\boldsymbol{x})$ will have a large value.

The number M of basis function is important:

- With too **few** basis functions, we **cannot separate** the data
- With too **many** basis functions, we will **overfit** the data

The RBF Layer

Basis Functions



The number of basis functions depends of the initialization method:

• Max Like for Gauss (MLG):	C basis functions
• K-means (KM):	k basis functions
• In-class K-means (ICKM):	C * k basis functions

Basis functions can initialized using any or all initialization methods.

When there are C = 10 classes, we can have M = 85 basis functions if we have:

- 10 MLG
- 25 KM, where k = 25
- 50 ICKM, where k = 5

Linear Discriminant Layer

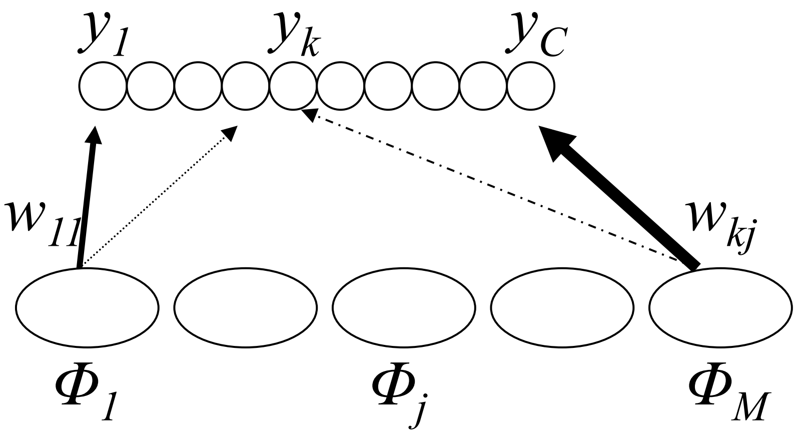
Each output node y_k is a weighted sum (i.e. linear combination) of the basis function outputs:

$$y_k(\boldsymbol{x}) = \sum_{j=1}^{M} w_{kj}\phi_j(\boldsymbol{x}) + w_{k,bias}$$

To learn a optimal weights (***W***), we minimize the sum-of-squared error function using our labeled training set:

$$E = \frac{1}{2}\sum_{n}\sum_{k}(y_k(\boldsymbol{x}^n) - t_k^n)^2$$

Here, t_k^n is the *k*-th target of the *n*-th training example: 1 if x^n has label k and 0 otherwise.



Outputs:

Weights:

Basis Functions:

Linear Discriminant Layer

Good news: There is a closed-form solution to minimizing the sum-of-squares errors function.

Let $\boldsymbol{\Phi}$ be an N x M matrix of outputs from the M basis functions for the N data points.

Let $\boldsymbol{W}$ be a C x M matrix of weights from the M basis functions to the C output nodes.

Let $\boldsymbol{T}$ be a N x C matrix of targets for the N data points.

The optimal sets of weights is found using the ‘pseudo-inverse’ solution:

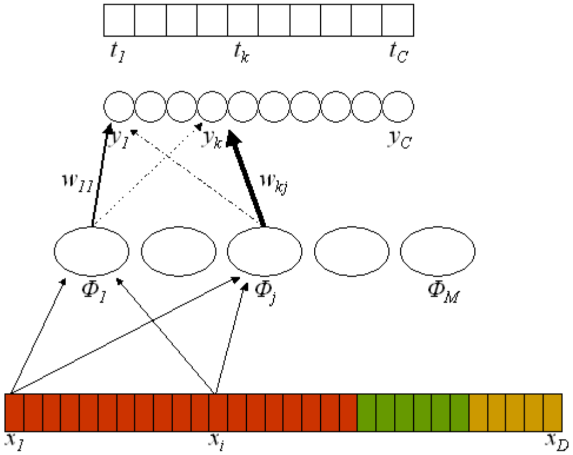
$$\boldsymbol{W}^{\mathrm{T}} = (\boldsymbol{\Phi}^{T}\ \boldsymbol{\Phi})^{-1}\ \boldsymbol{\Phi}^{T}\boldsymbol{T}$$

Finding the optimal weights, given fixed parameters for the RBFs, is fast.

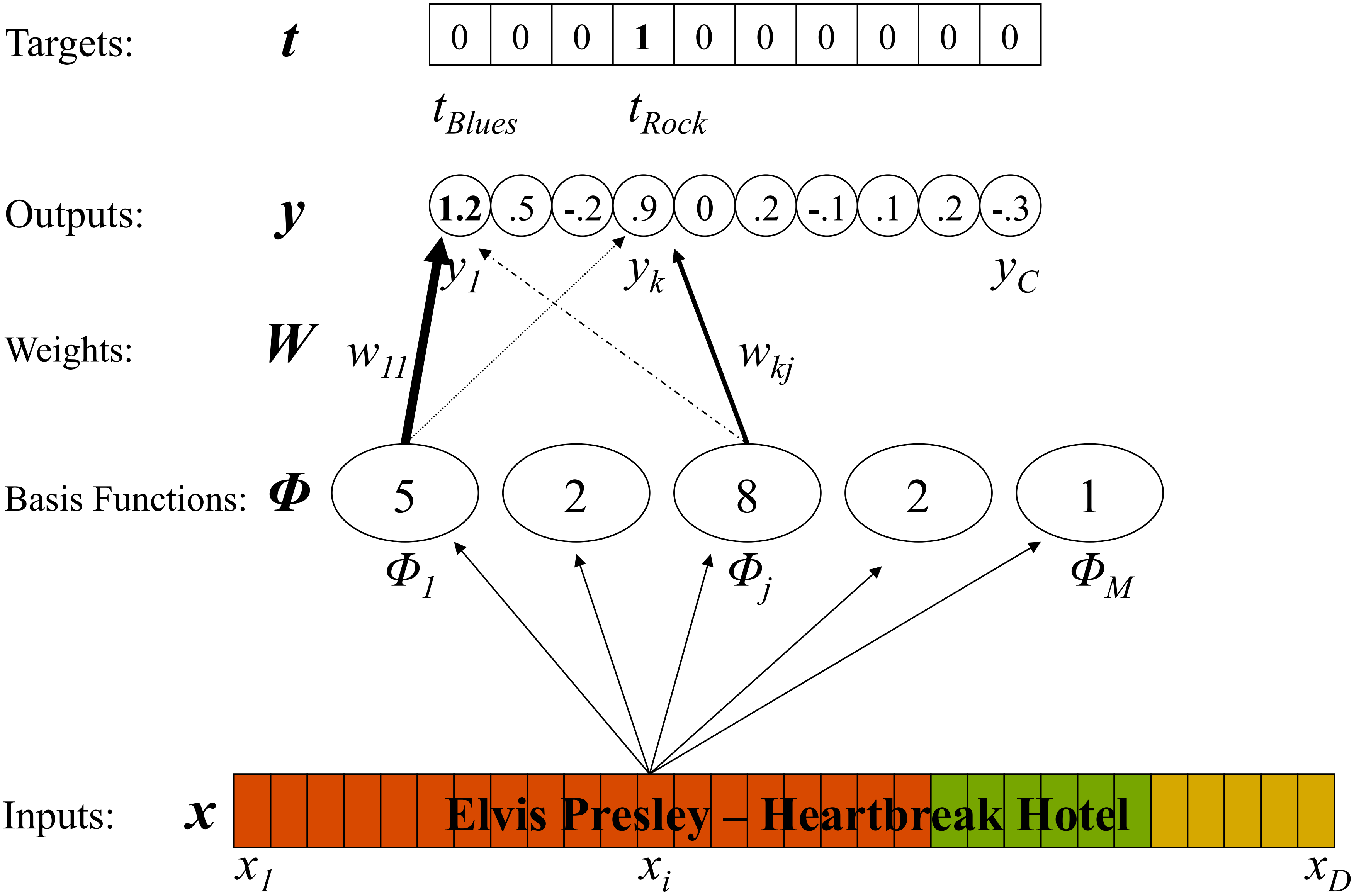
- 3 matrix multiplications
- 1 matrix inversion

Summary

1. **Number of basis functions**
   - Depends on initialization methods
2. **Initializing parameters to basis functions -** (μ , σ).
   - **Unsupervised** K-means clustering (**KM**)
   - **Supervised** Maximum Likelihood for Gaussian (**MLG**), In-class K-means clustering (**ICKM**)
   - **Combine above methods together**
3. **Compute Optimal Parameters for Linear Discriminants**



A (Misclassification) Example:



Targets: $\boldsymbol{t}$

0	0	0	**1**	0	0	0	0	0	0

t_{Blues} t_{Rock}

Outputs: $\boldsymbol{y}$

1.2	.5	-.2	.9	0	.2	-.1	.1	.2	-.3

y_1 y_k y_C

Weights: $\boldsymbol{W}$

w_{11} w_{kj}

Basis Functions: $\boldsymbol{\Phi}$

5 2 8 2 1

Φ_1 Φ_j Φ_M

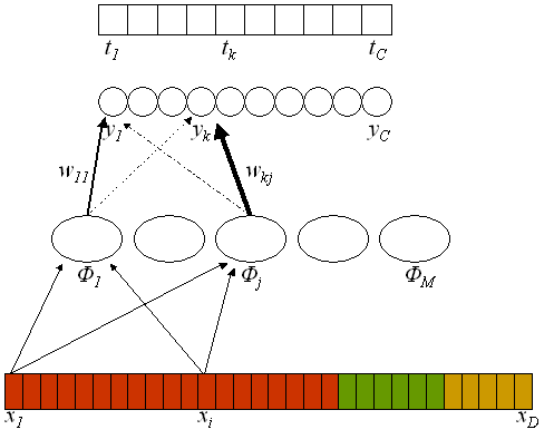
Inputs: $\boldsymbol{x}$

Elvis Presley – Heartbreak Hotel

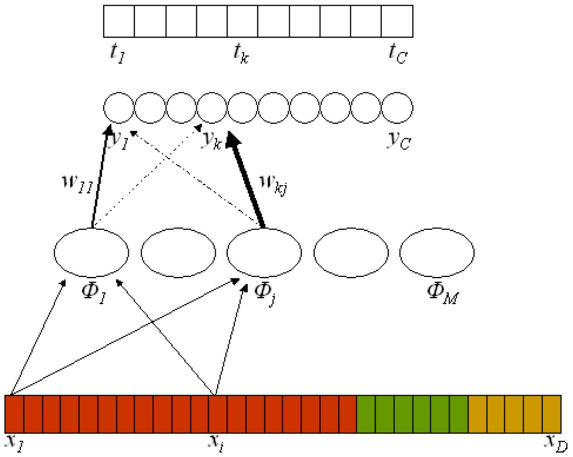
x_1 x_i x_D

Summary

1. **Number of basis functions**
   - Depends on initialization methods
2. **Initializing parameters to basis functions -** (μ , σ)
   - Three Initialization Methods – KM, MLG, ICKM
3. **Compute Optimal Parameters for Linear Discriminants**
4. **Improving parameters of the basis functions -** (μ , σ).
   - **Gradient Descent**



Gradient Descent on μ , σ



We differentiate our error function

$$E = \frac{1}{2} \sum_n \sum_k (y_k(\boldsymbol{x}^n) - t_k^n)^2$$

with respect to σ_j and m_{ji}

$$\frac{\partial E}{\partial \sigma_j}(x^n) = \sum_k \{y_k(\boldsymbol{x}^n) - t_k^n\} w_{kj} exp\{-\frac{||\boldsymbol{x}^n - \boldsymbol{\mu}^j||^2}{2\sigma_j^2}\} \frac{||\boldsymbol{x}^n - \boldsymbol{\mu}^j||^2}{\sigma_j^3}$$

$$\frac{\partial E}{\partial \mu_{ji}}(x^n) = \sum_k \{y_k(\boldsymbol{x}^n) - t_k^n\} w_{kj} exp\{-\frac{||\boldsymbol{x}^n - \boldsymbol{\mu}_j||^2}{2\sigma_j^2}\} \frac{(x_i^n - \mu_{ji})}{\sigma_j^2}$$

We then update σ_j and $m_{ji:}$ by moving down the error surface:

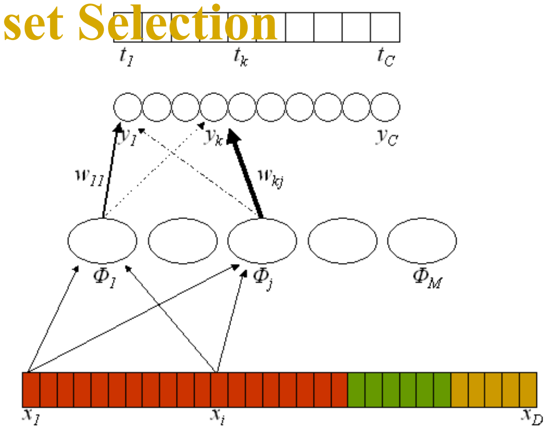
$$\sigma_j \leftarrow \sigma_j - \eta_1 \frac{\partial E}{\partial \sigma_j}$$

$$\mu_{ji} \leftarrow \mu_{ji} - \eta_2 \frac{\partial E}{\partial \mu_{ji}}$$

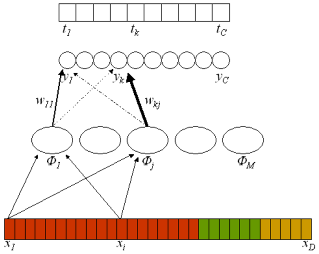
The learning rate scale factors, η_1 and η_2 , decrease each epoch.

Summary

1. **Number of basis functions – *M***
   - Depends on initialization methods, gradient descent
2. **Initializing parameters to basis functions -** (μ , σ)
   - Three Initialization Methods – KM, MLG, ICKM
3. **Compute Optimal Parameters for Linear Discriminants**
4. **Improving parameters of the basis functions -** (μ , σ).
   - Gradient Descent
5. **Further Reduce Dimensionality of input vector - *x***
   - **Feature Subset Selection**



Feature Subset Selection



This can be particularly useful when there are redundant and/or noisy features.

We will use the Forward Stepwise Selection algorithm to pick a good subset of features.

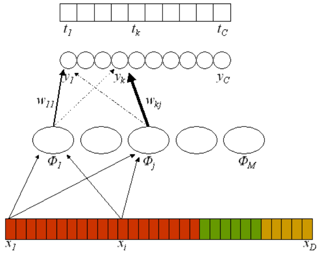
$S^d \leftarrow$ set of *d* selected features
$U \leftarrow$ set of remaining features

$$S^{d+1} = S^d \cup \{\operatorname{argmax}_{f \in U} \text{accuracy}(\{f \cup S^d\})\}$$

Here accuracy() is the classification accuracy of an RBF network that is trained using the set of $\{f \cup S^d\}$ features.

This algorithm requires that we train $(D^2 / 2)$ networks, where D is the dimension of the feature vector.

A Quick Review:



Input space: Audio Tracks preprocessed into $\{\text{Real}\}^D$

Output space: Musical Genre = {0,…,9)

Training Set: 1000 Human Labeled Audio Tracks

Classifier: Radial Basis (RBF) Networks

The parameter of the radial basis function network are:

M – the number of basis functions

($\boldsymbol{\mu}_j$, σ_j) – parameters for j-th basis function

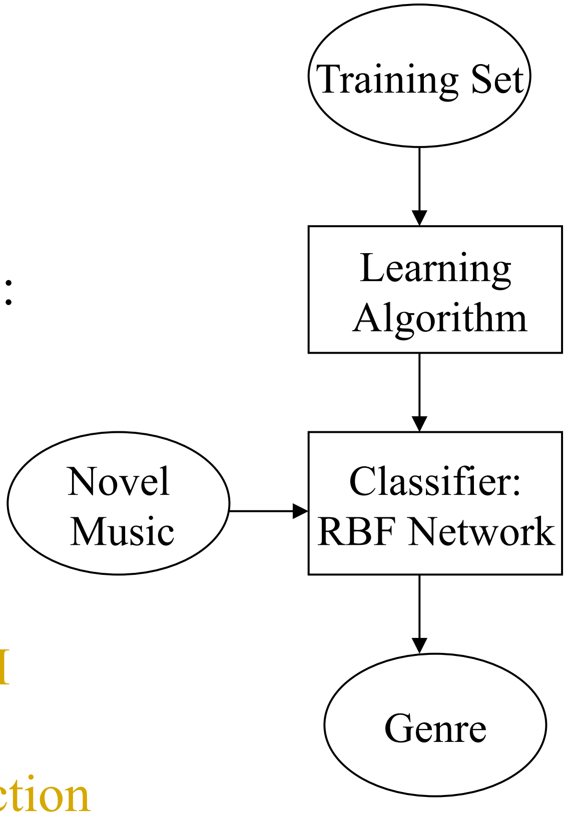
D – the dimension of the input feature vector

Our decision include:

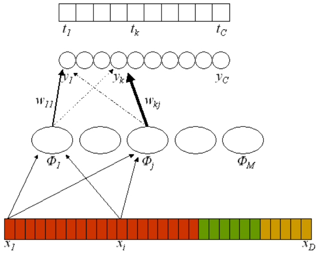
Which initialization method? – KM, MLG, ICKM

Whether to use Gradient Descent?

Which features to include? – Feature Subset Selection



Results



Experimental Setup

30 second clips from 1000 songs covering 10 genres

30 dimensional feature vectors are extracted from each sample

10-Fold cross-validation of randomly permuted data

For each fold, we divide the data into a

- 800 song training set
- 100 song hold-out set – to prevent overfitting during gradient descent
- 100 song test set

Baseline Comparison (random guessing) = 0.10

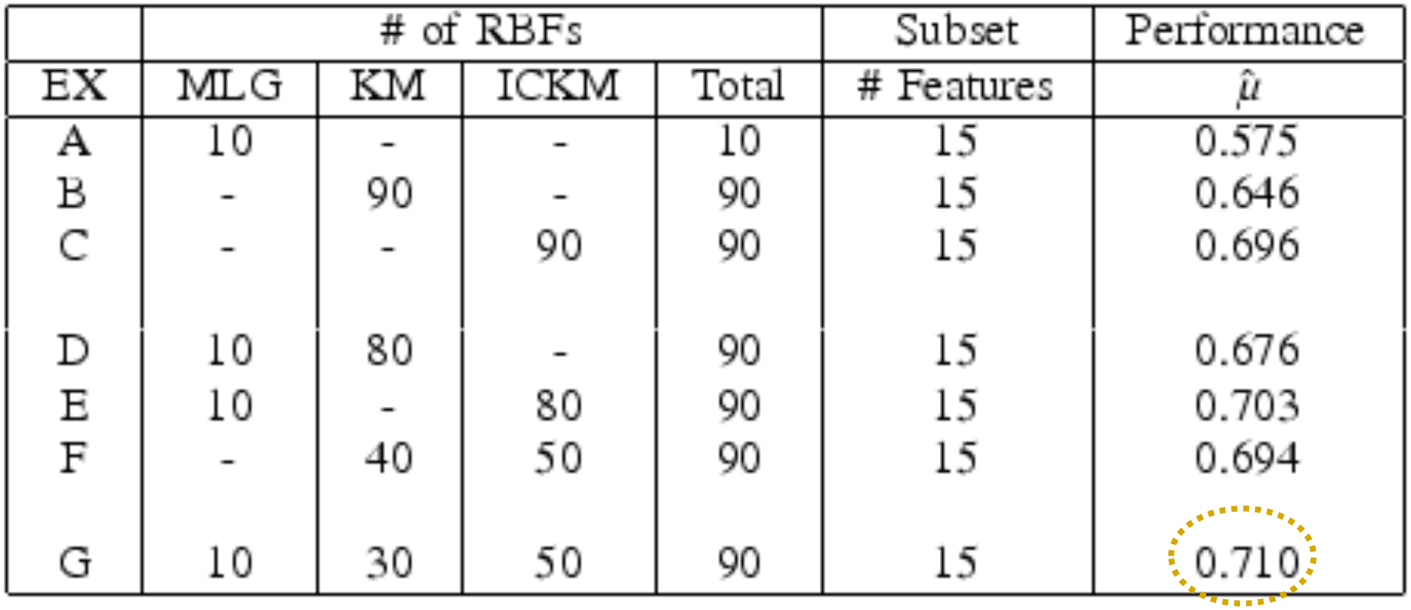
Classification Accuracy is

# correctly classified / 1000

Significance in defined by

2 * sqrt(1000*0.5*0.5) / 1000 ≈ **0.03**

Initialization Methods

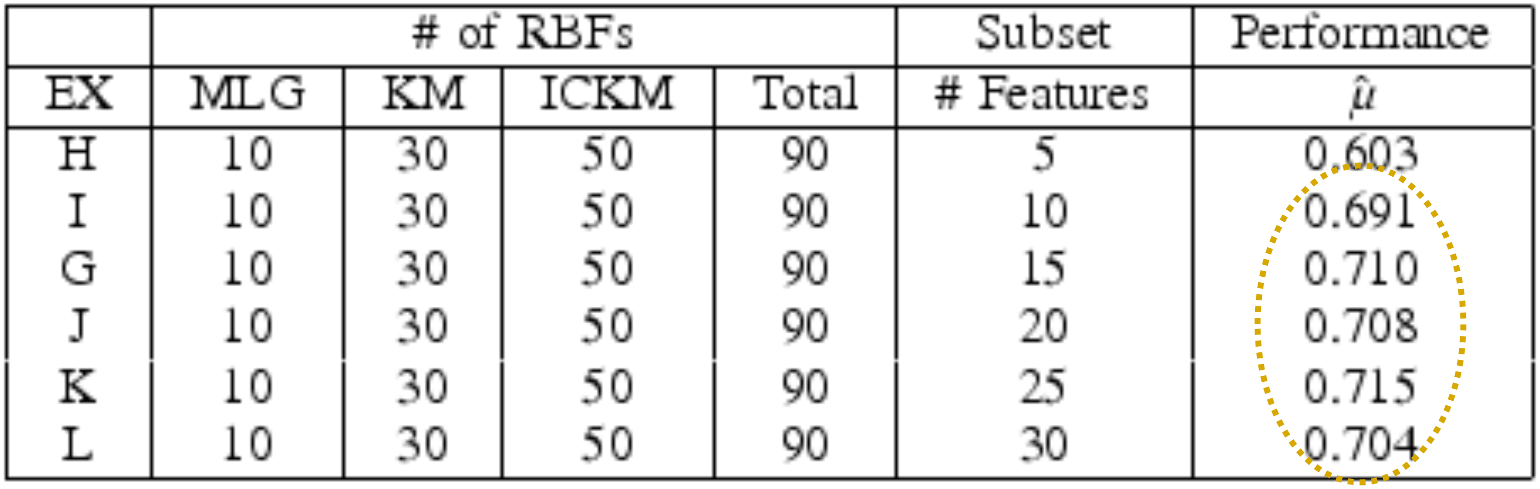


	# of RBFs				Subset	Performance
EX	MLG	KM	ICKM	Total	# Features	$\hat{\mu}$
A	10	-	-	10	15	0.575
B	-	90	-	90	15	0.646
C	-	-	90	90	15	0.696
D	10	80	-	90	15	0.676
E	10	-	80	90	15	0.703
F	-	40	50	90	15	0.694
G	10	30	50	90	15	0.710

Observation

1. **Multiple initialization method produces better classification than using only one initialization method.**

Feature Subset Selection



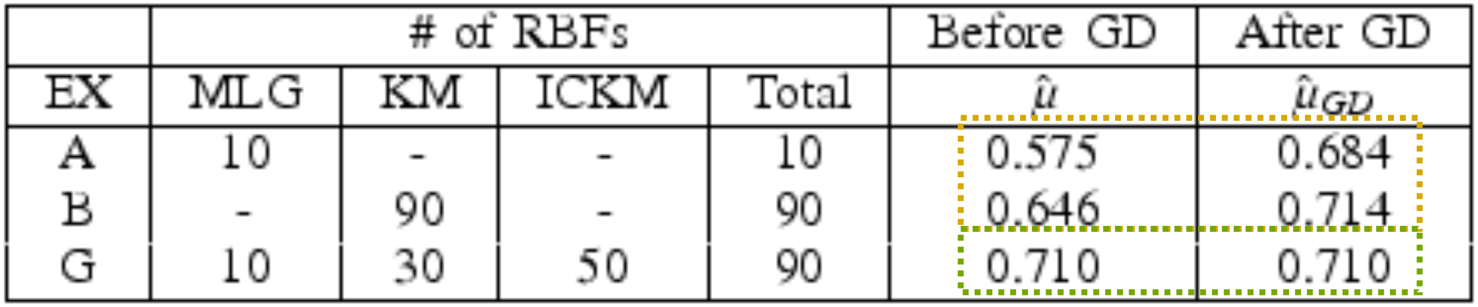
	# of RBFs				Subset	Performance
EX	MLG	KM	ICKM	Total	# Features	$\hat{\mu}$
H	10	30	50	90	5	0.603
I	10	30	50	90	10	0.691
G	10	30	50	90	15	0.710
J	10	30	50	90	20	0.708
K	10	30	50	90	25	0.715
L	10	30	50	90	30	0.704

Observations

1. **Including more than 10 good features does not significantly improve classification results.**

2. **Features selected by FSS algorithm include timbral, rhythmic content and pitch content features**

Initialization Methods



	# of RBFs				Before GD	After GD
EX	MLG	KM	ICKM	Total	$\hat{u}$	$\hat{u}_{GD}$
A	10	-	-	10	0.575	0.684
B	-	90	-	90	0.646	0.714
G	10	30	50	90	0.710	0.710

Observations

1. **Gradient Descent boosts performance for when initialization methods do a poor job.**
2. **Gradient descent does NOT help when a combination of initialization methods produce good classification results.**

Comparison with Previous Results

RBF networks:

71% (std 1.5%)

Human classification in similar experiment (Tzanetakis & Cook 2001):

70%

GMM with 3 Gaussians per class (Tzanetakis & Cook 2001):

61% (std 4%)

Support Vector Machine (SVM) (Li & Tzanetakis 2003):

69.1% (std 5.3%)

Linear Discriminant Analysis (LDA) (Li & Tzanetakis 2003):

71.1% (std 7.3%)

Why RBF Networks are a Natural Classifier

1. **Supervised and Unsupervised Learning Methods**
   - **Elvis Example**

2. **Fast Classification of Novel Music**
   - **Common property of many classifiers**

3. **Fast Training of Network**
   - **Combine multiple initialization methods**
   - **Closed-Form Linear Discriminants Calculation**

4. **Cognitively Plausible**
   - **Uses multiple stages of filtering to extract higher level information from lower level primitives.**

Why RBF Networks are a Natural Classifier

5. Allows for Flexible Classification System
    1. RBF networks can allow for a non-mutually exclusive classification.
    2. RBF networks can handle variable strengths of assignment.

Heartbreak Hotel can be given a target of 0.6 for blues and 0.7 for Rock.

Working with Musicologists to construct a more comprehensive classification system, and then collecting a data set the represents this system will be a valuable next step.

Relationship with Computer Vision Research

The is a closely coupled relationship between computer vision and computer audition.

- Both involve a high-dimensional digital medium.
- Common tasks include
  - Segmenting the digital medium
  - Classifying the digital medium
  - Retrieving the related examples from large databases

In the case of digital video, the two media are tied together.

A good AI multimedia system will rely on **Vision** and **Audition**.

Relationship with Computer Vision Research

Larger features Sets and Feature Subset Selection

One technique that has been successful in computer vision research is to automatically extract tens of thousands of features and then use features subset selection for find a small set (~30) of good features.

Computer Vision Features

1. Select sub-images of different sizes an locations
2. Alter resolution and scale factors.
3. Apply filters (e.g. Gabor filters)

Computer Audition Analog

1. Select sound samples of different lengths and starting locations
2. Alter pitches and tempos within the frequency domain
3. Apply filters (e.g. comb filters)

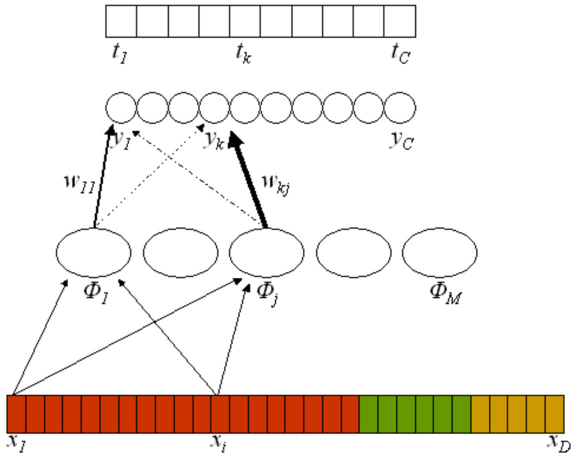
Future Work

1. Exploring More Flexible Labeling Systems
    - Non-mutually exclusive
    - Partial (Soft) Assignments
2. Segmenting of Music
    - Vector of Vectors representation
3. Incorporating Unlabeled Audio
    - Using EM algorithm

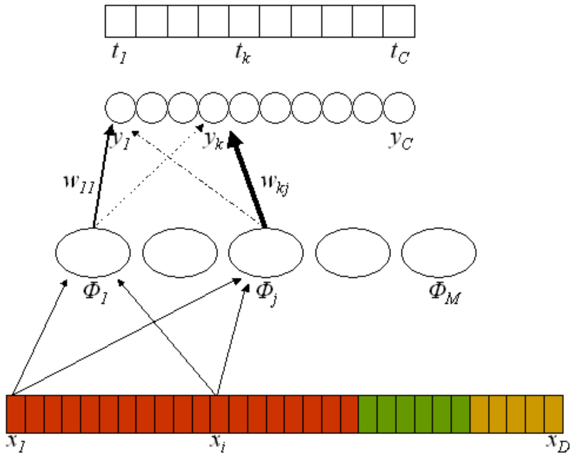
Questions for CRCA Music Group

1. Should we be considering other features?
   - Measure dynamic of a piece – Prof. Dubnov's Spectral Anticipations

2. Can you think of other application for mined audio content?
   - Multimedia Search Engine
   - "Hit Song Science"



The End



The motivation for this work came from undergraduate research under Perry Cook and George Tzanetakis in 2000-2001. Some of background information and code used in the development of RBF network is related to work on a project with Neil Alldrin and Andrew Smith during a Neural Networks course led by Prof Gary Cottrell in the Winter of 2003. This paper was develop with the help of Prof. Charles Elkan during his course on Machine Learning in the Spring of 2003. Prof. Gerald Balzano of the UCSD music department also provided helpful input with regards to some to the psycoacoustic ideas that are found in this paper.

Tzanetakis, G. & Cook, P.R. (2002) Musical Classification of Audio Signals. *IEEE Transactions on Speech and Audio Processing* 10(5)

Tzanetakis, G. & Li, T. (2003) Factors in Automatic Musical Classification of Audio Signals. *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*

Ullman, S., Vidal-Naquet, M. & Sali, E. (2002) Visual Features of Intermediate Complexity and their use in Classification. *Nature Neuroscience* 5(7):682-687.