Speech Pattern Recognition

• Soft pattern classification plus temporal sequence integration

• Supervised pattern classification: class labels used in training

• Unsupervised pattern classification: class labels not available or used
LECTURE ON PATTERN RECOGNITION

Pattern Extraction

Feature Vector

\[
\begin{bmatrix}
x_1 \\
x_2 \\
x_d
\end{bmatrix}
\]

Classification

\( \omega_k \)

\( 1 \leq k < K \)
• Training: learning parameters of classifier

• Testing: classify independent test set, compare with labels and score
Lecture on Pattern Recognition

F1 vs F2 for Pure Vowels /a/, /ae/, /i/, and /u/

- "data.a" ●
- "data.ae" +
- "data.i" □
- "data.u" ×
Feature Extraction Criteria

• Class discrimination
• Generalization
• Parsimony (efficiency)
plosive + vowel energies for 2 different gains
\[
\frac{\partial}{\partial t} \log CE(t) = \frac{\partial}{\partial t} (\log C + \log E(t)) \\
= \frac{\partial}{\partial t} \log E(t)
\]
Feature Vector Size

- Best representations for discrimination on training set are large (highly dimensioned)
- Best representations for generalization to test set are (typically) succinct
Dimensionality Reduction

• Principal components (i.e., SVD, KL transform, eigenanalysis ...)

• Linear Discriminant Analysis (LDA)

• Application-specific knowledge

• Feature Selection via PR Evaluation
FISHER’S LINEAR DISCRIMINANT

Projection of samples onto a line.
PR Methods

• Minimum Distance
• Discriminant Functions
• Linear Discriminant
• Nonlinear Discriminant
  (e.g., quadratic, neural networks)
• Statistical Discriminant Functions
Minimum Distance

- Vector or matrix representing element
- Define a distance function
- Choose the class of stored element closest to new input
- Choice of distance equivalent to implicit statistical assumptions
- For speech, temporal variability complicates this
\[ z_i = \text{template vector (prototype)} \]

\[ x = \text{input vector} \]

Choose \( i \) to minimize distance

\[
\arg_i \min \sqrt{(x - z_i)^T(x - z_i)} = \arg_i \min (x - z_i)^T(x - z_i) = \arg_i \min (x^T x + z_i^T z_i - 2 x^T z_i)
\]

\[
\arg_i \max \left( \frac{z_i^T z_i - 2 x^T z_i}{-2} \right) = \arg_i \max \left( x^T z_i - \frac{1}{2} z_i^T z_i \right)
\]

If \( z_i^T z_i = 1 \) for all \( i \) \( \Rightarrow \arg_i \max (x^T z_i) \)
Problems with Min Distance

• Proper scaling of dimensions (size, discrimination)
• For high dim, sparsely sampled space
Decision Rule for Min Distance

• Nearest Neighbor (NN) - in the limit of infinite samples, at most twice the error of optimum classifier

• k-Nearest Neighbor (kNN)

• Lots of storage for large problems; potentially large searches
Some Opinions

• Better to throw away bad data than to reduce its weight

• Dimensionality-reduction based on variance often a bad choice for supervised pattern recognition
**Discriminant Analysis**

- Discriminant functions max for correct class, min for others

- Decision surface between classes

- Linear decision surface for 2-dim is line, for 3 is plane; generally called hyperplane

- For 2 classes, surface at $\omega^T x + \omega_0 = 0$

- 2-class quadratic case, surface at $x^T W x + \omega^T x + \omega_0 = 0$
Pattern input

Function 1

Decision Process (e.g., max)

Function 2

Class choice
Training Discriminant Functions

• Minimum distance
• Fisher linear discriminant
• Gradient learning
Generalized Discriminators - ANNs

• McCulloch Pitts neural model
• Rosenblatt Perceptron
• Multilayer Systems
The Perceptron

McCulloch-Pitts Neuron - Rosenblatt Perceptron
**Perceptron Convergence**

If classes are linearly separable the following rule will converge in a finite number of steps:

For each pattern \( x \) at time step \( k \):

\[
\begin{cases}
    x(k) \in \text{class 1, } \omega^T(k)x(k) \leq 0 \\
    \Rightarrow \omega(k + 1) = \omega(k) + cx(k) \\
    x(k) \in \text{class 2, } \omega^T(k)x(k) \geq 0 \\
    \Rightarrow \omega(k + 1) = \omega(k) - cx(k) \\
\end{cases}
\]

\[
\text{else} \quad \omega(k + 1) = """" \omega(k)
\]
Multilayer Perceptrons

• Heterogeneous, “hard” nonlinearity: (DAID, 1961)

• Homogeneous, “soft” nonlinearity

(“modern” MLP)
The diagram represents a pattern recognition model. It consists of input features $x_1, x_2, \ldots, x_n$, each weighted by $w_{1j}, w_{2j}, \ldots, w_{nj}$, respectively. These weighted inputs are summed up by the symbol $\sum$, resulting in an output $y_j$. The output then passes through a function $f(y_j)$.
\[ f(y) = \frac{1}{1 + e^{-y}} \quad \text{(sigmoid)} \]

\[ 0 < f(y) < 1 \]
Some PR Issues

• Testing on the training set

• Training on the test set

• No. parameters vs no. training examples: overfitting and overtraining