SYDE 372
Introduction to Pattern Recognition

Distance Measures for Pattern Classification: Part I

Alexander Wong

Department of Systems Design Engineering
University of Waterloo

Outline

Distance measures for pattern classification

- Intuitively, two patterns that are sufficiently similar should be assigned to the same class.
- But what does “similar” mean?
  - How similar are these patterns quantitatively?
  - How similar are they to a particular class quantitatively?
- Since we represent patterns quantitatively as vectors in a feature space, it is often possible to:
  - use some measure of similarity between two patterns to quantify how similar their attributes are
  - use some measure of similarity between a pattern and a *prototype* to quantify how similar it is with a class

Minimum Euclidean Distance (MED) Classifier

- Definition:

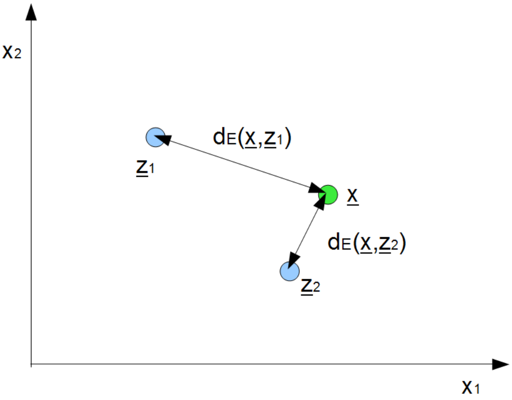
$$\underline{x} \in c_k \text{ iff } d_E(\underline{x}, \underline{z}_k) < d_E(\underline{x}, \underline{z}_l) \quad (1)$$

for all $l \neq k$, where

$$d_E(\underline{x}, \underline{z}_k) = [(\underline{x} - \underline{z}_k)^T (\underline{x} - \underline{z}_k)]^{1/2} \quad (2)$$

- Meaning: $\underline{x}$ belongs to class k if and only if the Euclidean distance between $\underline{x}$ and the prototype of c_k is less than the distance between $\underline{x}$ and all other class prototypes.

MED Classifier: Visualization



MED Classifier: Discriminant Function

- Simplifying the decision criteria $d_E(\underline{x}, \underline{z}_k) < d_E(\underline{x}, \underline{z}_l)$ gives us:

$$-\underline{z}_1{}^T \underline{x} + \frac{1}{2}\underline{z}_1{}^T \underline{z}_1 < -\underline{z}_2{}^T \underline{x} + \frac{1}{2}\underline{z}_2{}^T \underline{z}_2 \quad (3)$$

- This gives us the discrimination/decision function:

$$g_k(\underline{x}) = -\underline{z}_k{}^T \underline{x} + \frac{1}{2}\underline{z}_k{}^T \underline{z}_k \quad (4)$$

- Therefore, MED classification made based on **minimum** discriminant for given $\underline{x}$:

$$\underline{x} \in c_k \text{ iff } g_k(\underline{x}) < g_l(\underline{x}) \; \forall l \neq k \quad (5)$$

MED Classifier: Decision Boundary

- Formed by features equidistant to two classes ($g_k(\underline{x}) = g_l(\underline{x})$):

$$g(\underline{x}) = g_k(\underline{x}) - g_l(\underline{x}) = 0 \tag{6}$$

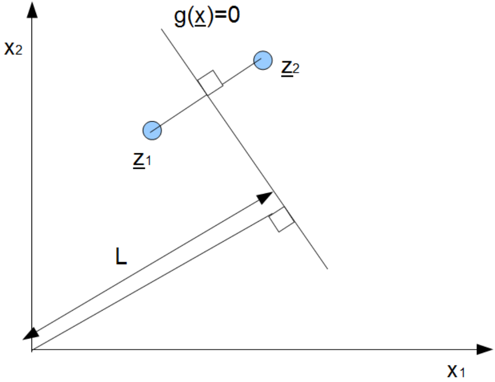
- For MED classifier, the **decision boundary** becomes:

$$g(\underline{x}) = (\underline{z}_k - \underline{z}_l)^T \underline{x} + \frac{1}{2}(\underline{z}_l{}^T \underline{z}_l - \underline{z}_k{}^T \underline{z}_k) = 0 \tag{7}$$

$$g(\underline{x}) = \underline{w}^T \underline{x} + w_o = 0 \tag{8}$$

MED Classifier: Decision Boundary Visualization

- The MED decision boundary is just a hyperplane with normal vector $\underline{w}$, a distance $\left|\frac{w_o}{|\underline{w}|}\right|$ from the origin



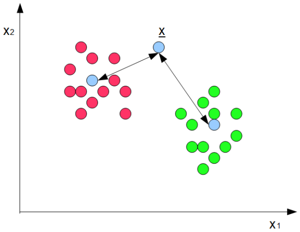
Prototype Selection

- So far, we've assumed that we have a specific prototype $\underline{z}_i$ for each class.
- But how do we select such a class prototype?
- Choice of class prototype will affect the way the classifier works
- Let us study the classification problem where:
  - We have a set of samples with known classes c_k
  - We need to determine a class prototype based on these labeled samples

Common prototypes: Sample Mean

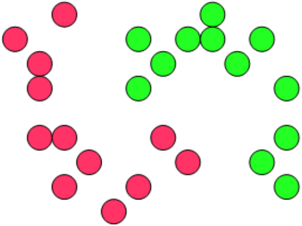
$$z_k(x) = \frac{1}{N_k} \sum_{i=1}^{N_k} \underline{x}_i \tag{9}$$

where N_k is the number of samples in class c_k and $\underline{x}_i$ is the i^{th} sample of c_k.



Common prototypes: Sample Mean

- **Advantages:**
  - + Less sensitive to noise and outliers
- **Disadvantages:**
  - - Poor at handling long, thin, tendril-like clusters



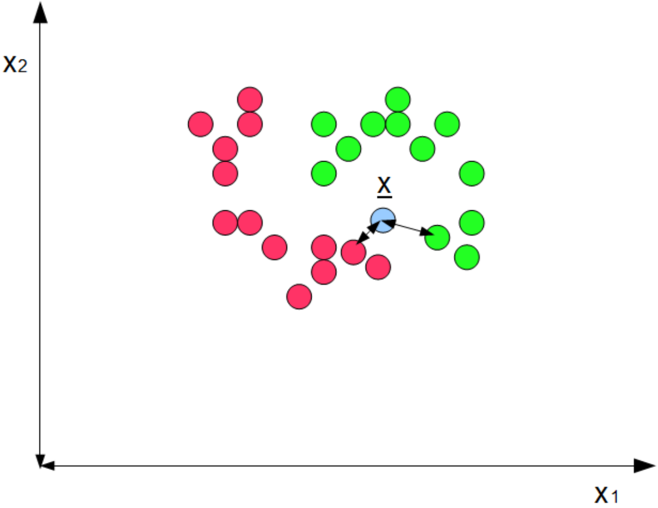
Common prototypes: Nearest Neighbor (NN)

- Definition:

$$z_k(x) = \underline{x}_k \text{ such that } d_E(\underline{x}, \underline{x}_k) = \min_i d_E(\underline{x}, \underline{x}_i) \;\; \forall \underline{x}_i \in c_k. \tag{10}$$

- Meaning: For a given $\underline{x}$ you wish to classify, you compute the distance between $\underline{x}$ and all labeled samples, and you assign $\underline{x}$ the same class as its nearest neighbor.

Common prototypes: Nearest Neighbor (NN)



Common prototypes: Nearest Neighbor (NN)

- **Advantages:**
  - + Better at handling long, thin, tendril-like clusters
- **Disadvantages:**
  - - More sensitive to noise and outliers
  - - Computationally complex (need to re-compute all prototypes for each new point)

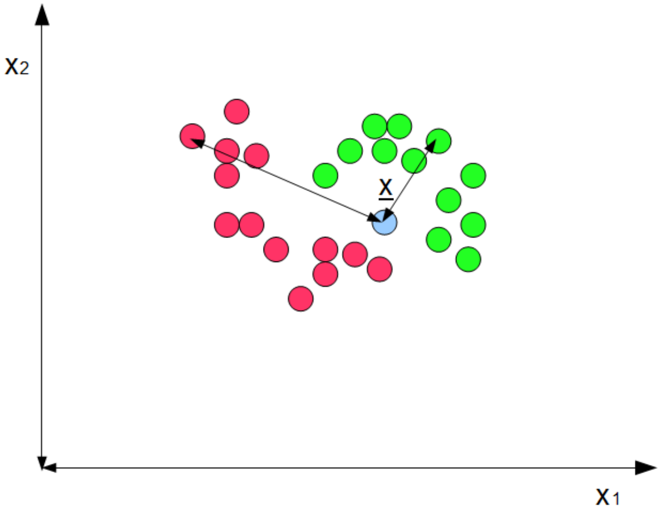
Common prototypes: Furthest Neighbor (FNN)

- Definition:

$$z_k(x) = \underline{x}_k \text{ such that } d_E(\underline{x}, \underline{x}_k) = \max_i d_E(\underline{x}, \underline{x}_i) \;\; \forall \underline{x}_i \in c_k. \tag{11}$$

- Meaning: For a given $\underline{x}$ you wish to classify, you compute the distance between $\underline{x}$ and all labeled samples, and you define the prototype in each cluster as that point furthest from $\underline{x}$.

Common prototypes: Furthest Neighbor (FNN)



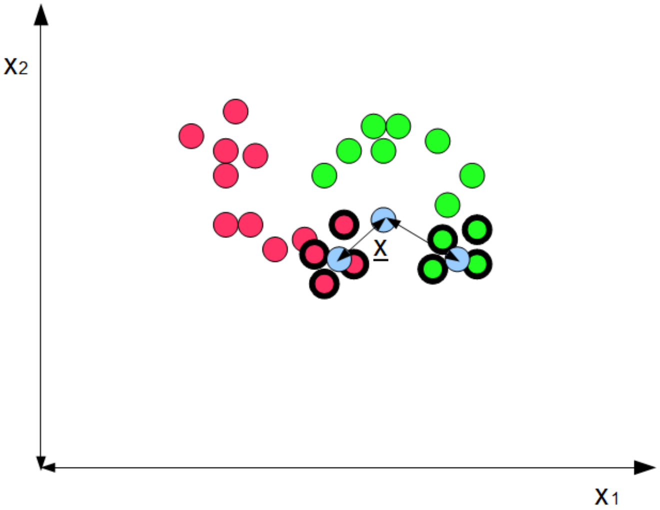
Common prototypes: Furthest Neighbor (FNN)

- **Advantages:**
  - + More tight, compact clusters
- **Disadvantages:**
  - - More sensitive to noise and outliers
  - - Computationally complex (need to re-compute all prototypes for each new point)

Common prototypes: K-nearest Neighbor

- Idea:
  - Nearest neighbor is sensitive to noise, but handles long, tendril-like clusters well
  - Sample mean is less sensitive to noise, but poorly handles long, tendril-like clusters
  - What if we combine the two ideas?
- Definition: For a given $\underline{x}$ you wish to classify, you compute the distance between $\underline{x}$ and all labeled samples, and you define the prototype in each cluster as the *samplemean* of the K samples within that cluster that is nearest $\underline{x}$.

Common prototypes: K-nearest Neighbor

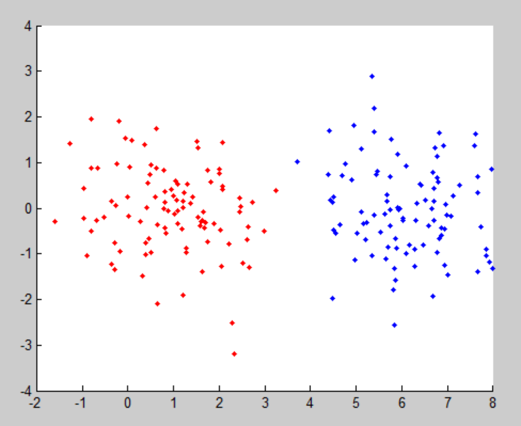


Common prototypes: K-nearest Neighbor

- **Advantages:**
  - + Less sensitive to noise and outliers
  - + Better at handling long, thin, tendril-like clusters
- **Disadvantages:**
  - - Computationally complex (need to re-compute all prototypes for each new point)

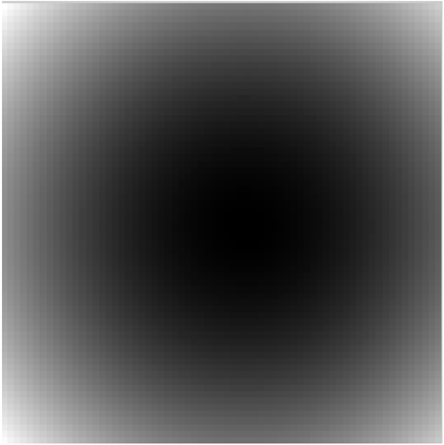
Example: Performance from using different prototypes

- Features are Gaussian in nature, different means, uncorrelated, equal variant:



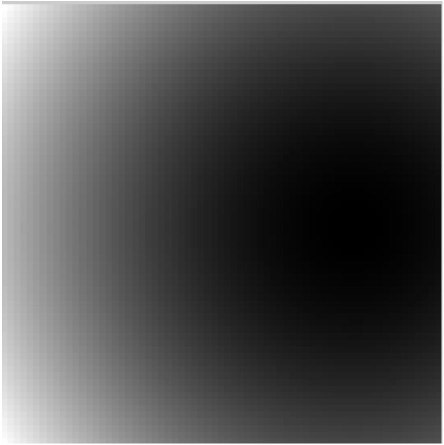
Example: Performance from using different prototypes

- Euclidean distance from sample mean for class A:



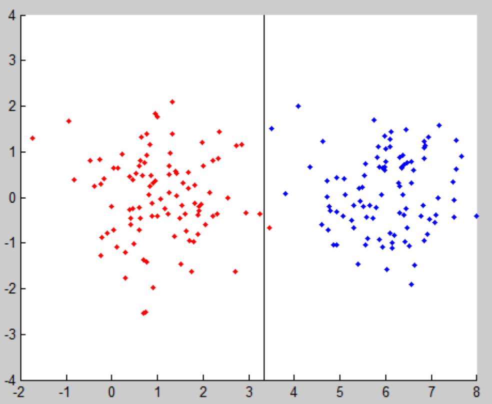
Example: Performance from using different prototypes

- Euclidean distance from sample mean for class B:



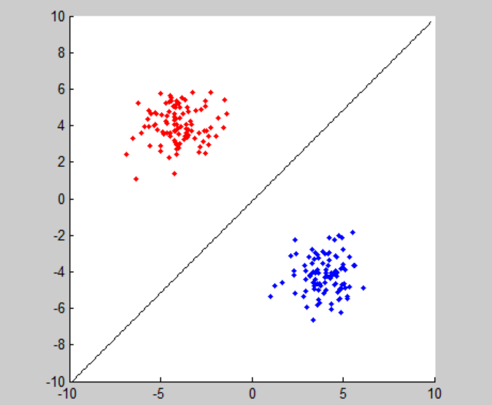
Example: Performance from using different prototypes

- MED decision boundary



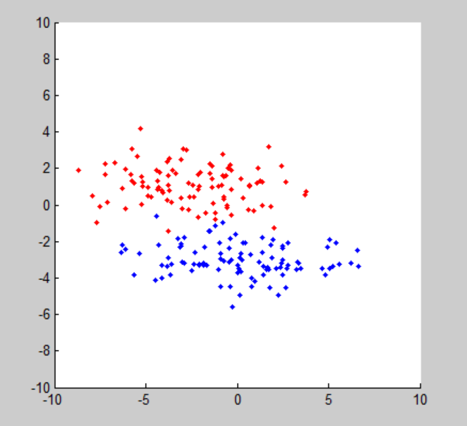
Example: Performance from using different prototypes

- Features are Gaussian in nature, different means, uncorrelated, equal variant:



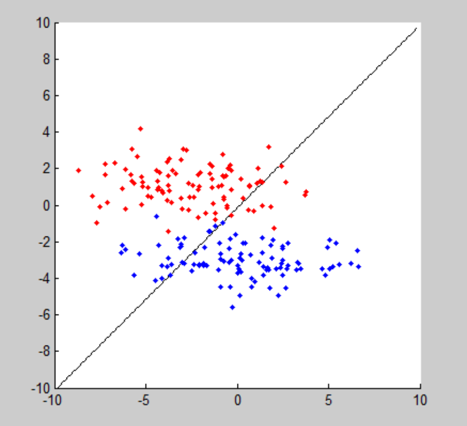
Example: Performance from using different prototypes

- Features are Gaussian in nature, different means, uncorrelated, different variances:



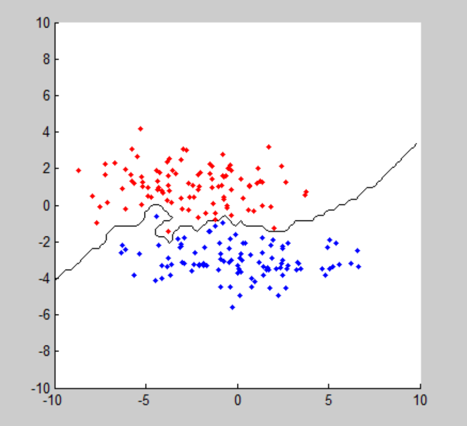
Example: Performance from using different prototypes

- MED decision boundary:



Example: Performance from using different prototypes

- NN decision boundary:



MED Classifier: Example

- Suppose we are given the following labeled samples:
  - Class 1: $\underline{x}_1 = [2 \;\; 1]^T, \underline{x}_2 = [3 \;\; 2]^T, \underline{x}_3 = [2 \;\; 7]^T, \underline{x}_4 = [5 \;\; 2]^T$.
  - Class 2: $\underline{x}_1 = [3 \;\; 3]^T, \underline{x}_2 = [4 \;\; 4]^T, \underline{x}_3 = [3 \;\; 9]^T, \underline{x}_4 = [6 \;\; 4]^T$.
- Suppose we wish to build a MED classifier using sample means as prototypes.
  - Compute the discriminate function for each class.
  - Sketch the decision boundary.

MED Classifier: Example

- Step 1: Find sample mean prototypes for each class:

$$z_1 = \frac{1}{N_1}\sum_{i=1}^{N_1} \underline{x}_i$$
$$\begin{aligned} \underline{z}_1 &= \tfrac{1}{4}\left\{[2\ \ 1]^T + [3\ \ 2]^T + [2\ \ 7]^T + [5\ \ 2]^T\right\} \\ \underline{z}_1 &= \tfrac{1}{4}\left\{[12\ \ 12]^T\right\} \\ \underline{z}_1 &= [3\ \ 3]^T. \end{aligned} \tag{12}$$

$$z_2 = \frac{1}{N_2}\sum_{i=1}^{N_2} \underline{x}_i$$
$$\begin{aligned} \underline{z}_2 &= \tfrac{1}{4}\left\{[3\ \ 3]^T + [4\ \ 4]^T + [3\ \ 9]^T + [6\ \ 4]^T\right\} \\ \underline{z}_1 &= \tfrac{1}{4}\left\{[16\ \ 20]^T\right\} \\ \underline{z}_2 &= [4\ \ 5]^T. \end{aligned} \tag{13}$$

MED Classifier: Example

- Step 2: Find discriminant functions for each class based on MED decision rule:
- Recall that the MED decision criteria for the two class case is:

$$d_E(\underline{x}, \underline{z}_1) < d_E(\underline{x}, \underline{z}_2) \quad (14)$$

$$[(\underline{x} - \underline{z}_1)^T(\underline{x} - \underline{z}_1)]^{1/2} < [(\underline{x} - \underline{z}_2)^T(\underline{x} - \underline{z}_2)]^{1/2} \quad (15)$$

$$(\underline{x} - \underline{z}_1)^T(\underline{x} - \underline{z}_1) < (\underline{x} - \underline{z}_2)^T(\underline{x} - \underline{z}_2) \quad (16)$$

$$-\underline{z}_1{}^T\underline{x} + \frac{1}{2}\underline{z}_1{}^T\underline{z}_1 < -\underline{z}_2{}^T\underline{x} + \frac{1}{2}\underline{z}_2{}^T\underline{z}_2 \quad (17)$$

MED Classifier: Example

- Plugging in z_1 and z_2 gives us:

$$-\underline{z}_1{}^T\underline{x} + \frac{1}{2}\underline{z}_1{}^T\underline{z}_1 < -\underline{z}_2{}^T\underline{x} + \frac{1}{2}\underline{z}_2{}^T\underline{z}_2 \qquad (18)$$

$$\begin{aligned} &-[3\ \ 3]^{T^T}[x_1\ \ x_2]^T + \tfrac{1}{2}[3\ \ 3]^{T^T}[3\ \ 3]^T \\ &< -[4\ \ 5]^{T^T}[x_1\ \ x_2]^T + \tfrac{1}{2}[4\ \ 5]^{T^T}[4\ \ 5]^T \end{aligned} \qquad (19)$$

$$-[3\ \ 3][x_1\ \ x_2]^T + \frac{1}{2}[3\ \ 3][3\ \ 3]^T < -[4\ \ 5][x_1\ \ x_2]^T + \frac{1}{2}[4\ \ 5][4\ \ 5]^T \qquad (20)$$

MED Classifier: Example

- Plugging in z_1 and z_2 gives us:

$$-3x_1 - 3x_2 + 9 < -4x_1 - 5x_2 + \frac{41}{2} \tag{21}$$

- Therefore, the discriminant functions are:

$$g_1(x_1, x_2) = -3x_1 - 3x_2 + 9 \tag{22}$$

$$g_2(x_1, x_2) = -4x_1 - 5x_2 + \frac{41}{2} \tag{23}$$

MED Classifier: Decision Boundary

- Step 3: Find decision boundary between classes 1 and 2
- For MED classifier, the decision boundary is

$$g(\underline{x}) = (\underline{z}_k - \underline{z}_l)^T \underline{x} + \frac{1}{2}(\underline{z}_l{}^T \underline{z}_l - \underline{z}_k{}^T \underline{z}_k) = 0 \qquad (24)$$

$$g(x_1, x_2) = g_1(x_1, x_2) - g_2(x_1, x_2) = 0. \qquad (25)$$

Plugging in the discriminant functions g_1 and g_2 gives us:

$$g(x_1, x_2) = -3x_1 - 3x_2 + 9 - (-4x_1 - 5x_2 + \frac{41}{2}) = 0 \quad (26)$$

MED Classifier: Decision Boundary

- Grouping terms:

$$-3x_1 - 3x_2 + 9 + 4x_1 + 5x_2 - \frac{41}{2} = 0 \tag{27}$$

$$2x_2 + x_1 - \frac{23}{2} = 0 \tag{28}$$

$$x_2 = -\frac{1}{2}x_1 + \frac{23}{4} \tag{29}$$

- Therefore, the decision boundary is just a straight line with a slope of $-\frac{1}{2}$ and an offset of $\frac{23}{4}$.

MED Classifier: Decision Boundary

- Step 4: Sketch decision boundary