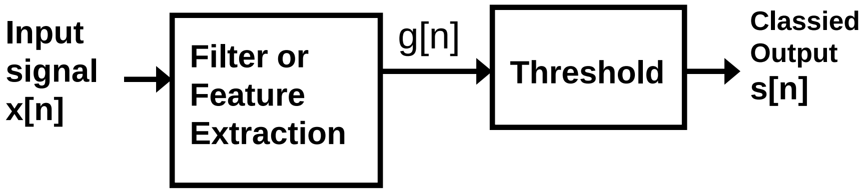
ECGR 6114
Classifiers

Classifiers

- Gaussian optimum threshold
- Multivariate Gaussian
- Bayesian Classifier
- Linear Discriminant Classifier
- K-Means Classifier
- K-Nearest Neighbor Classifier
- Minimum-Distance Classifier
- Supervised / Unsupervised

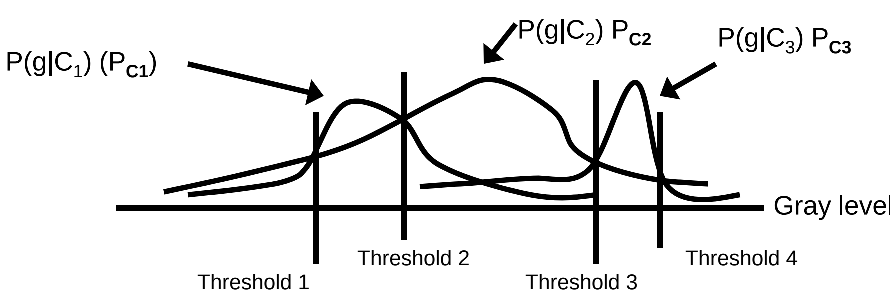
Single-Feature (Scalar) Threshold Classifier

- Suppose a signal processing system with 1 feature
- Scalar Feature g



Optimum Single-Feature Threshold

- Optimum Bayes classifier : use maximimum likelihood ratio
  - $P(g|C_1)$ = probability density in region of class C_1
  - P_{C1} = *a priori* probability of class C_1
  - g = observed scalar feature value
  - Choose class $s[n]=C_\alpha$ when
    $P(g|C_\alpha)\, P_{C\,\alpha} \geq P(g|C_n)\, P_{C\,n}$ for all C_n (choose most likely)



Gaussian pdf

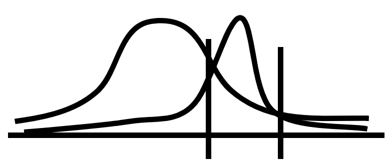
- Features are often well-modeled by Gaussian
- Gaussian pdf :

$$p(g \mid C_\alpha) = \frac{1}{\sigma_\alpha \sqrt{2\pi}} e^{-(g-\mu_\alpha)^2 / 2(\sigma_\alpha)^2}$$

- For class C_α with mean μ_α and standard deviation σ_α
- Bayes optimum classifier:
  - Choose class C_α when
    $P(g|C_\alpha)\ P_{C\,\alpha} \geq P(g|C_n)\ P_{C\,n}$ for all C_n (choose most likely)

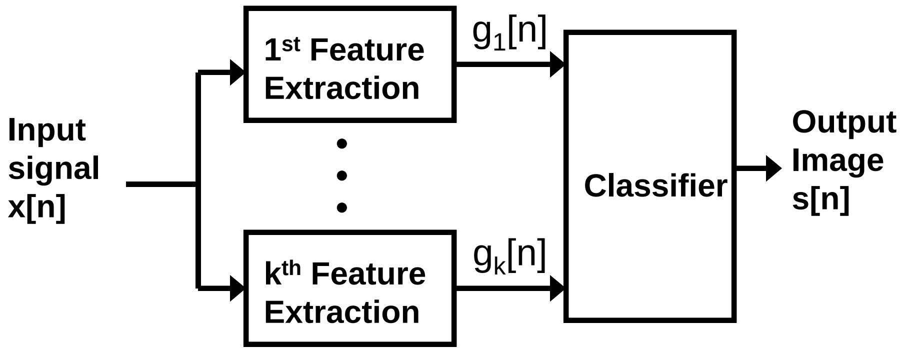
Gaussian Optimal Threshold

- For a single-feature system, the output g[n] is a scalar
- Note that for Gaussian pdf:
  If $\ln(P_{C\alpha}/\sigma_\alpha) - (g - \mu_\alpha)^2/2(\sigma_\alpha)^2 \geq \ln(P_{Cn}/\sigma_n) - (g - \mu_n)^2/2(\sigma_n)^2$
  Then $P(g|C_\alpha)\ P_{C\alpha} \geq P(g|C_n)\ P_{Cn}$
- So, can classify based on exponent
- 2-class Bayes optimum Gaussian scalar classifier, solve:
  $\ln(P_{C1}/\sigma_1) - (g - \mu_1)^2/2(\sigma_1)^2 = \ln(P_{C2}/\sigma_2) - (g - \mu_2)^2/2(\sigma_2)^2$
- Quadratic, so 2 solutions = 2 thresholds



Vector-Classifier

- Suppose a signal processing system with k features



N-dimensional Feature Vector $g = \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_N \end{bmatrix}$

Optimum Vector-Feature "Threshold"

- Optimum Bayes classifier : use maximimum likelihood ratio again,
  - P(**g**|C1) = probability density in region of class C_1
  - P_{C1} = probability of class C_1
  - **g** = observed N-dimensional feature vector value
  - Choose class C_α when
    $P(\mathbf{g}|C_\alpha)\, P_{C\,\alpha} \geq P(\mathbf{g}|C_n)\, P_{C\,n}$ for all C_n (choose most likely)
- Same form as before, except feature is a feature vector
- For 2-features, thresholds are closed curves (circle, ellipse)
- For 3 features …. Etc.

Multivariate Gaussian pdf

- Features are often well-modeled by multivariate Gaussian for output vector **g** of dimension N
- Multivariate Gaussian pdf

$$p(g \mid C_\alpha) = \frac{1}{|\Sigma_\alpha|^{1/2}(2\pi)^{N/2}} e^{-(g-u_\alpha)^T(\Sigma_\alpha)^{-1}(g-u_\alpha)/2}$$

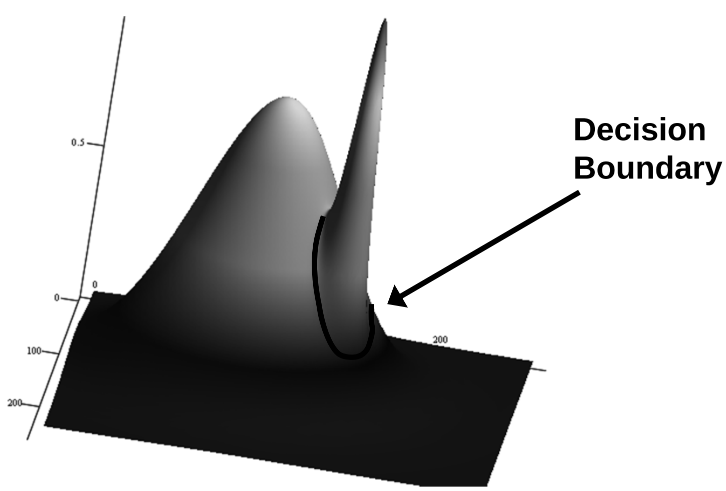
- For class C_α with mean vector μ_α = **E{ g }** and

  covariance matrix Σ_α **= E{ (g_α - μ_α) (g_α - μ_α)^T }**
- Bayes optimum classifier:
  - Choose class C_α when

    P(**g**|C_α) $P_{C\,\alpha}$ ≥ P(**g**|C_n) $P_{C\,n}$ for all C_n (choose most likely)
- In similar fashion, can use logarithmic form for classifier

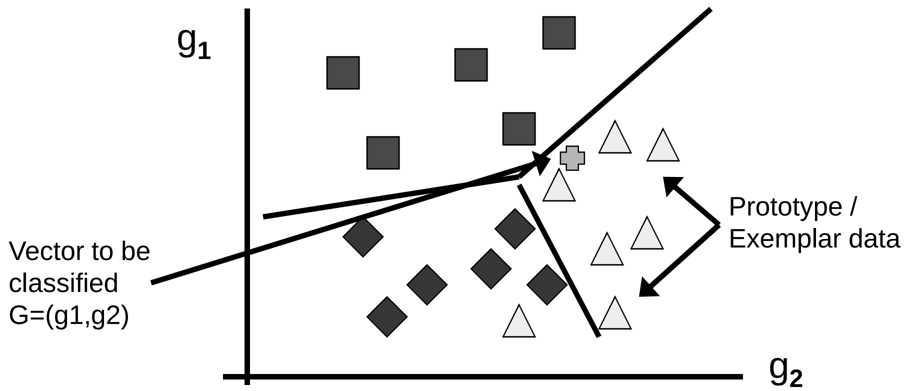
Multivariate Gaussian Example

- N=2 dimensional vector, 2 classes shown



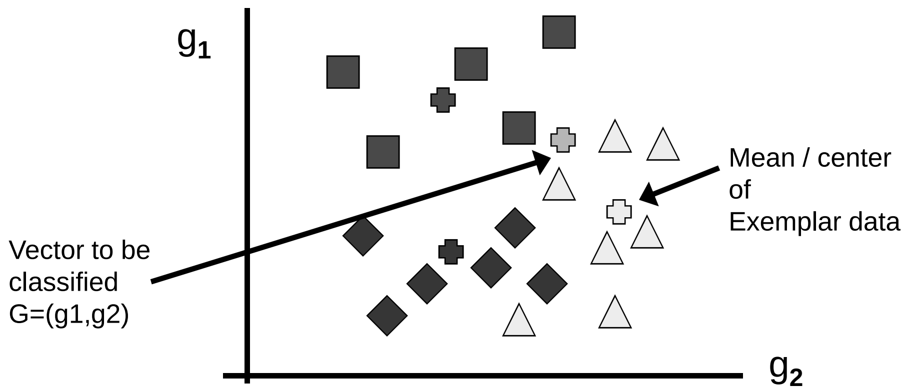
Linear Discriminant Classifier

- Simple clustering: Linear boundary (line or hyperplane)
- Distribution-free (non-parametric) method



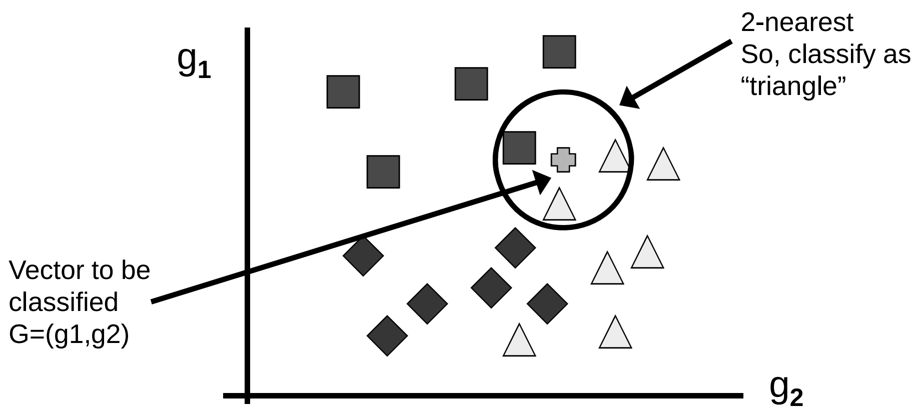
K-Means Classifier

- K-means clustering (minimum mean distance classifier): compute cluster centers, classifier assigns to class with closest center (mean)



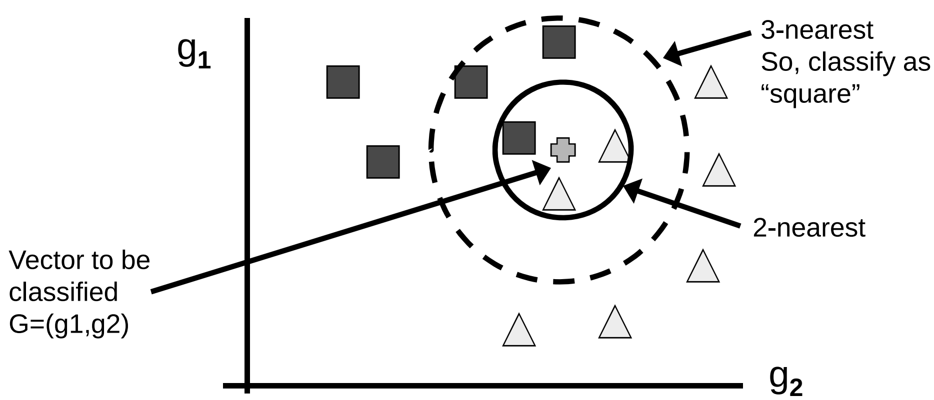
K-Nearest Neighbor Classifier

- K-nearest neighbor: find the k nearest neighbors of the vector G, assigns to majority class of the k neighbors



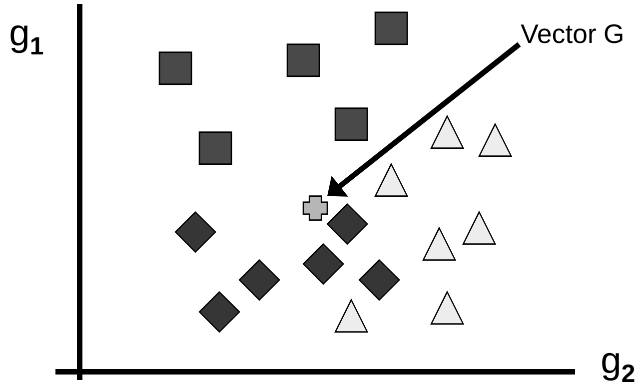
K-Nearest Neighbor Classifier

- K-nearest neighbor: find the k nearest neighbors of the vector G, assigns to majority class of the k neighbors



Minimum-Distance Classifier

- Minimum-distance Classifier: (1-nearest neighbor) find the nearest neighbor of the vector G, and assign to that class



Supervised/Unsupervised Classifiers

- Supervised Classification:
  – A set of examples/prototype/exemplar data fom each class are used to design the classifier
- Unsupervised Classification:
  – No set of examples/prototype/exemplar
  – Typically look for clusters/concentrations in feature space

Unsupervised Classifier

- As a first step, an unsupervised classifier would look at the data and organize it into a number of classes based on "clumping" of points in feature space
- Then, any of the prior methods, such as nearest neighbor, could be used to assign a vector to a class