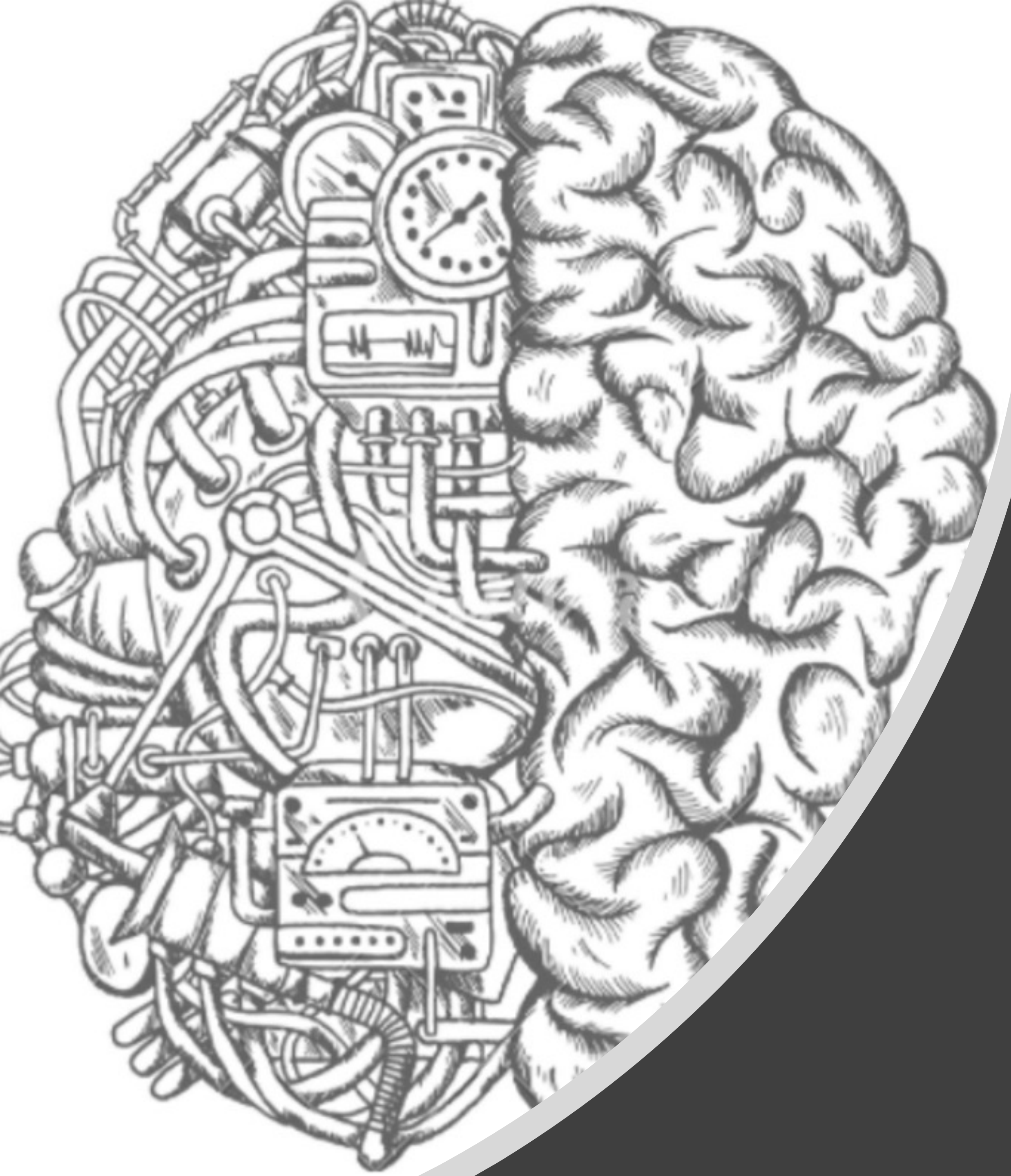


Machine Learning: Models and Applications

Lecture 6

Lecturer: Xinchao Wang

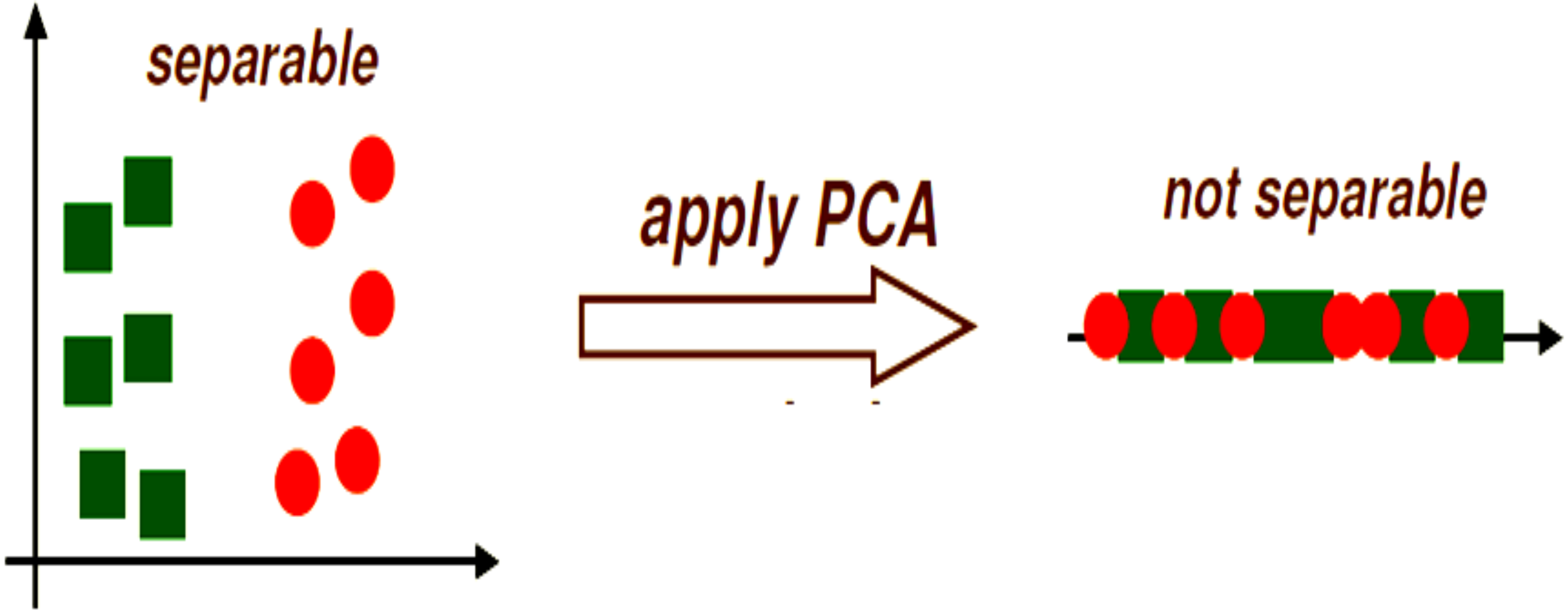
Overview

- Fisher Linear Discriminant (DHS Chapter 3 and notes based on course by Olga Veksler, Univ. of Western Ontario)
- Generative vs. Discriminative Classifiers
- Linear Discriminant Functions (notes based on Olga Veksler's)

Fisher Linear Discriminant Analysis (LDA/FDA/FLDA)

- PCA finds directions to project the data so that variance is maximized
- PCA does not consider *class labels*
- Variance maximization not necessarily beneficial for classification

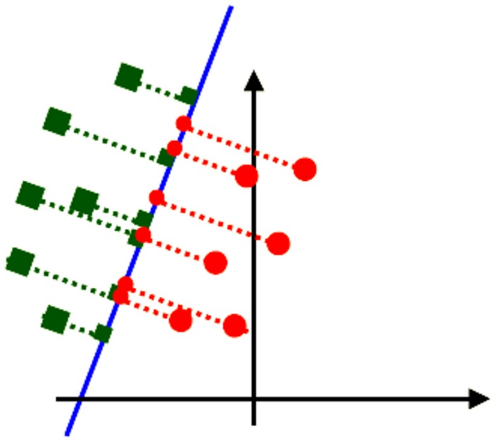
Data Representation vs. Data Classification



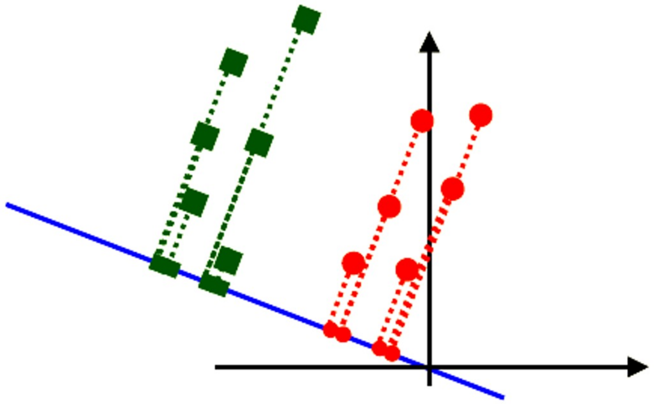
- Fisher Linear Discriminant: project to a line which preserves direction useful for *data classification*

Fisher Linear Discriminant

- Main idea: find projection to a line such that samples from different classes are well separated

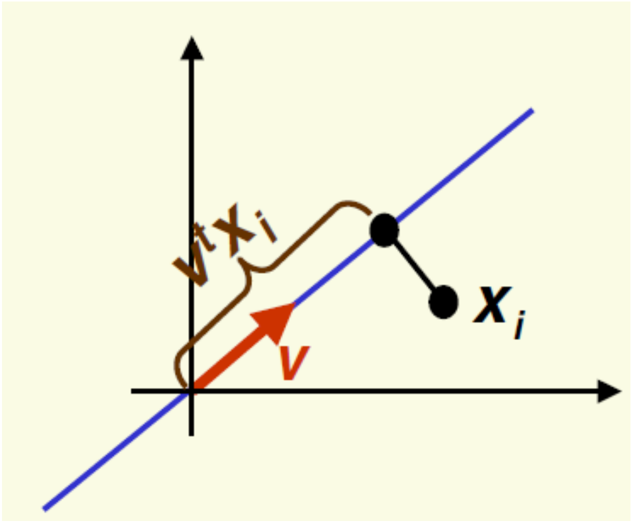


bad line to project to, classes are mixed up



good line to project to, classes are well separated

- Suppose we have 2 classes and d-dimensional samples $\mathbf{x}_1,\ldots,\mathbf{x}_n$ where:
  - **n_1 samples come from the first class**
  - **n_2 samples come from the second class**
- Consider projection on a line
- Let the line direction be given by unit vector **v**
- The scalar $\mathbf{v}^t\mathbf{x}_i$ is the distance of the projection of $\mathbf{x}_i$ from the origin
- Thus, $\mathbf{v}^t\mathbf{x}_i$ is the projection of $\mathbf{x}_i$ into a one dimensional subspace

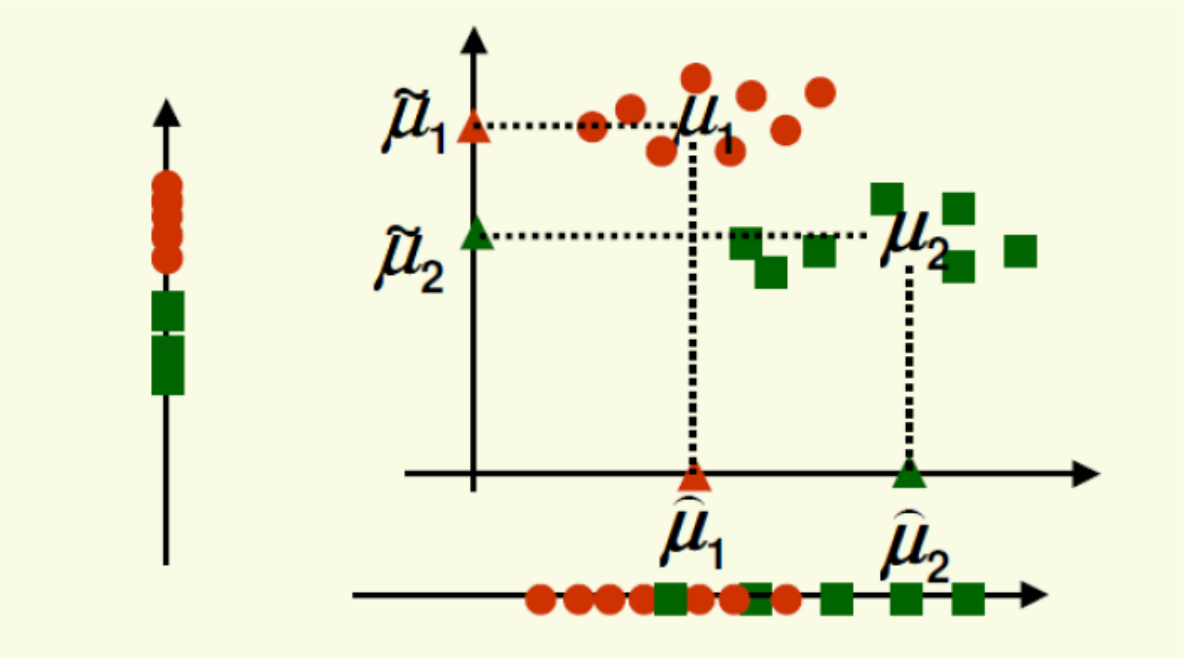


- The projection of sample $\mathbf{x}_i$ onto a line in direction $\mathbf{v}$ is given by $\mathbf{v}^t\mathbf{x}_i$
- How to measure separation between projections of different classes?
- Let $\tilde{\mu}_1$ and $\tilde{\mu}_2$ be the means of projections of classes 1 and 2
- Let μ_1 and μ_2 be the means of classes 1 and 2
- $|\tilde{\mu}_1 - \tilde{\mu}_2|$ seems like a good measure

$$\tilde{\mu}_1 = \frac{1}{n_1}\sum_{x_i \in C1}^{n_1} v^t x_i = v^t\left(\frac{1}{n_1}\sum_{x_i \in C1}^{n_1} x_i\right) = v^t \mu_1$$

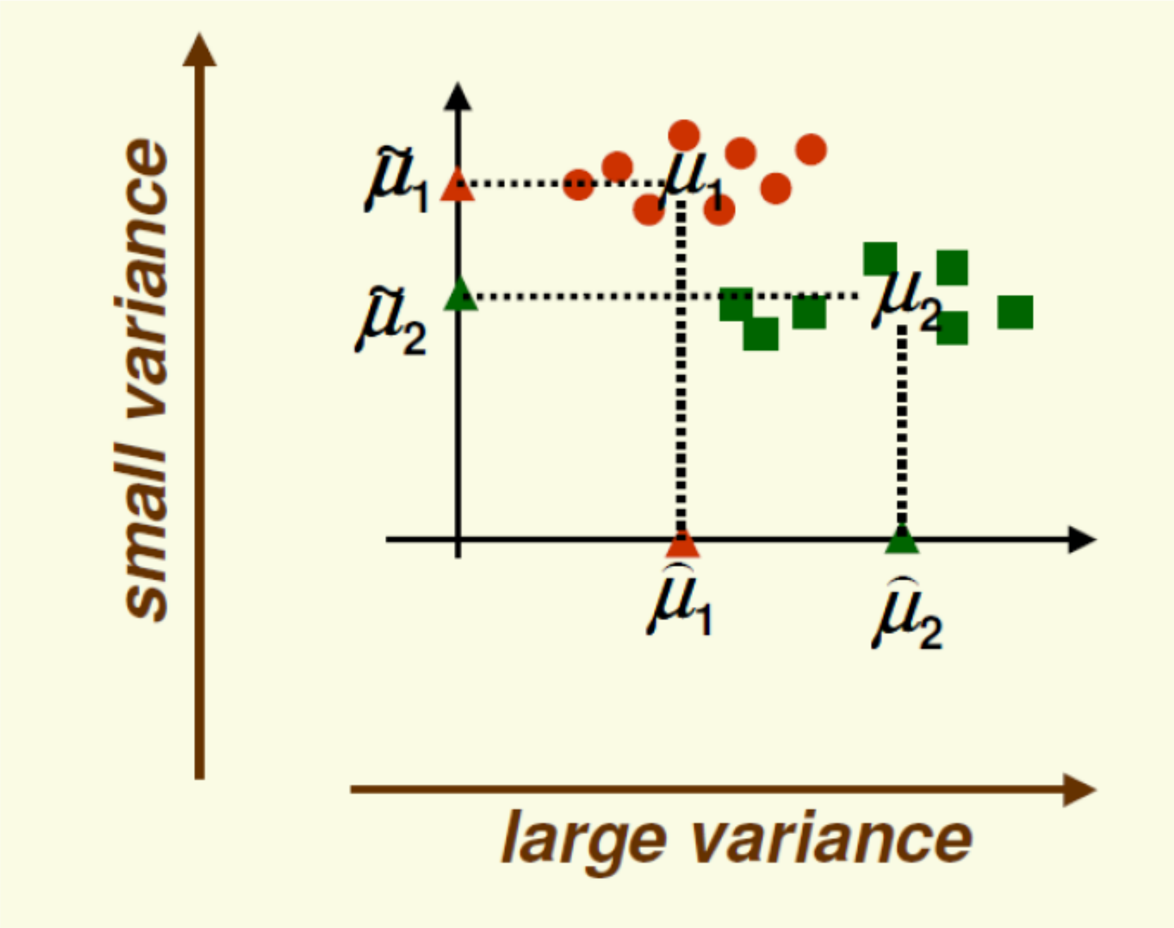
$$\textit{similarly}, \qquad \tilde{\mu}_2 = v^t \mu_2$$

- How good is $|\tilde{\mu}_1 - \tilde{\mu}_2|$ as a measure of separation?
  - The larger it is, the better the expected separation



- The vertical axis is a better line than the horizontal axis to project to for class separability
- However $|\tilde{\mu}_1 - \tilde{\mu}_2| < |\hat{\mu}_1 - \hat{\mu}_2|$

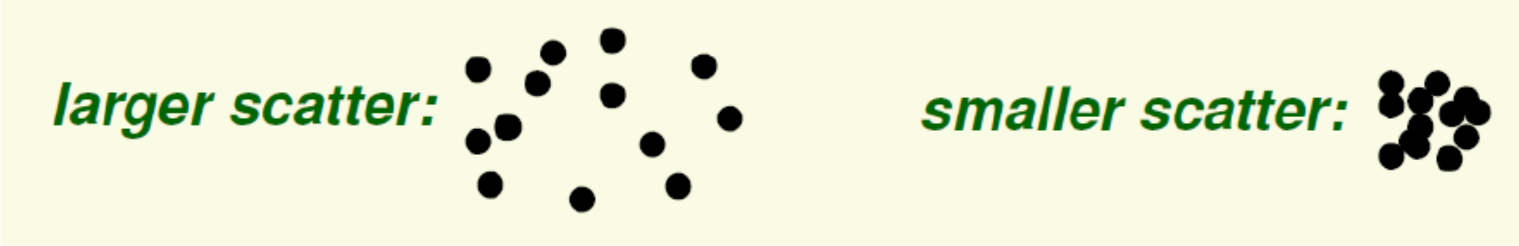
- The problem with $|\tilde{\mu}_1 - \tilde{\mu}_2|$ is that it does not consider the variance of the classes



- We need to normalize $|\tilde{\mu}_1 - \tilde{\mu}_2|$ by a factor which is proportional to variance
- For samples $\mathbf{z_1,\ldots,z_n}$, the sample mean is: $\mu_z = \frac{1}{n}\sum_{i=1}^{n} z_i$
- Define scatter as:

$$s = \sum_{i=1}^{n} (z_i - \mu_z)^2$$

- Thus scatter is just sample variance multiplied by **n**
  - Scatter measures the same thing as variance, the spread of data around the mean
  - Scatter is just on different scale than variance



- Fisher Solution: normalize $|\tilde{\mu}_1 - \tilde{\mu}_2|$ by scatter
- Let $\mathbf{y_i} = \mathbf{v^t x^i}$, be the projected samples
- The scatter for projected samples of class 1 is

$$\tilde{s}_1^2 = \sum_{y_i \in Class\ 1} (y_i - \tilde{\mu}_1)^2$$

- The scatter for projected samples of class 2 is

$$\tilde{s}_2^2 = \sum_{y_i \in Class\ 2} (y_i - \tilde{\mu}_2)^2$$

Fisher Linear Discriminant

- We need to normalize by both scatter of class 1 and scatter of class 2
- The Fisher linear discriminant is the projection on a line in the direction ***v*** which maximizes

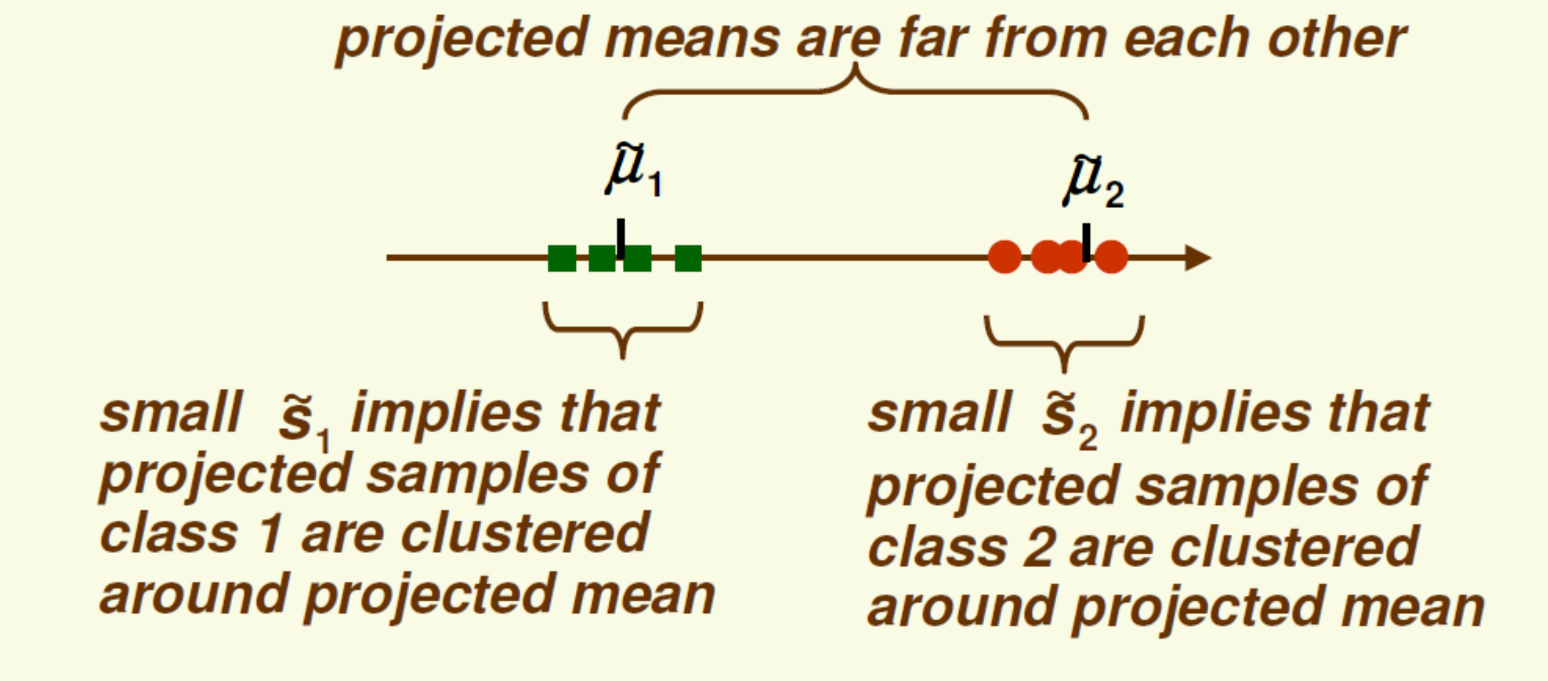
$$J(v) = \frac{\overbrace{(\tilde{\mu}_1 - \tilde{\mu}_2)^2}^{\textit{want projected means far from each other}}}{\tilde{\mathbf{s}}_1^2 + \tilde{\mathbf{s}}_2^2}$$

want scatter in class 1 is as small as possible, i.e. samples of class 1 cluster around the projected mean $\tilde{\mu}_1$

want scatter in class 2 is as small as possible, i.e. samples of class 2 cluster around the projected mean $\tilde{\mu}_2$

$$J(v) = \frac{(\tilde{\mu}_1 - \tilde{\mu}_2)^2}{\tilde{s}_1^2 + \tilde{s}_2^2}$$

- If we find **v** which makes **J(v)** large, we are guaranteed that the classes are well separated



Fisher Linear Discriminant - Derivation

$$J(v) = \frac{(\tilde{\mu}_1 - \tilde{\mu}_2)^2}{\tilde{s}_1^2 + \tilde{s}_2^2}$$

- All we need to do now is express **J(v)** as a function of **v** and maximize it
  - Straightforward but need linear algebra and calculus
- Define the class scatter matrices $\mathbf{S_1}$ and $\mathbf{S_2}$. These measure the scatter of original samples x_i (before projection)

$$S_1 = \sum_{x_i \in Class\ 1} (x_i - \mu_1)(x_i - \mu_1)^t$$
$$S_2 = \sum_{x_i \in Class\ 2} (x_i - \mu_2)(x_i - \mu_2)^t$$

- Define within class scatter matrix

$$S_W = S_1 + S_2$$

$$\tilde{s}_1^2 = \sum_{y_i \in Class\ 1} (y_i - \tilde{\mu}_1)^2$$

- $\boldsymbol{y_i = v^t x_i}$ and $\tilde{\mu}_1 = v^t \mu_1$

$$\begin{aligned}
\tilde{s}_1^2 &= \sum_{y_i \in Class\ 1} \left(v^t x_i - v^t \mu_1\right)^2 \\
&= \sum_{y_i \in Class\ 1} \left(v^t (x_i - \mu_1)\right)^t \left(v^t (x_i - \mu_1)\right) \\
&= \sum_{y_i \in Class\ 1} \left((x_i - \mu_1)^t v\right)^t \left((x_i - \mu_1)^t v\right) \\
&= \sum_{y_i \in Class\ 1} v^t (x_i - \mu_1)(x_i - \mu_1)^t v = v^t S_1 v
\end{aligned}$$

- Similarly

$$\tilde{s}_2^2 = \mathbf{v}^t \mathbf{S}_2 \mathbf{v}$$

$$\tilde{s}_1^2 + \tilde{s}_2^2 = \mathbf{v}^t \mathbf{S}_1 \mathbf{v} + \mathbf{v}^t \mathbf{S}_2 \mathbf{v} = \mathbf{v}^t \mathbf{S}_W \mathbf{v}$$

- Define between class scatter matrix

$$\mathbf{S}_B = (\mu_1 - \mu_2)(\mu_1 - \mu_2)^t$$

- $\boldsymbol{S_B}$ measures separation of the means of the two classes before projection
- The separation of the projected means can be written as

$$\begin{aligned}(\tilde{\mu}_1 - \tilde{\mu}_2)^2 &= \left(\mathbf{v}^t \mu_1 - \mathbf{v}^t \mu_2\right)^2 \\ &= \mathbf{v}^t (\mu_1 - \mu_2)(\mu_1 - \mu_2)^t \mathbf{v} \\ &= \mathbf{v}^t \mathbf{S}_B \mathbf{v}\end{aligned}$$

- Thus our objective function can be written:

$$J(v) = \frac{(\tilde{\mu}_1 - \tilde{\mu}_2)^2}{\tilde{s}_1^2 + \tilde{s}_2^2} = \frac{v^t S_B v}{v^t S_W v}$$

- Maximize **J(v)** by taking the derivative w.r.t. **v** and setting it to 0

$$\frac{d}{dv} J(v) = \frac{\left(\frac{d}{dv} v^t S_B v\right) v^t S_W v - \left(\frac{d}{dv} v^t S_W v\right) v^t S_B v}{\left(v^t S_W v\right)^2}$$

$$= \frac{(2S_B v) v^t S_W v - (2S_W v) v^t S_B v}{\left(v^t S_W v\right)^2} = 0$$

Need to solve $\quad \mathbf{v}^t S_W \mathbf{v}(S_B \mathbf{v}) - \mathbf{v}^t S_B \mathbf{v}(S_W \mathbf{v}) = 0$

$$\Rightarrow \frac{\mathbf{v}^t S_W \mathbf{v}(S_B \mathbf{v})}{\mathbf{v}^t S_W \mathbf{v}} - \frac{\mathbf{v}^t S_B \mathbf{v}(S_W \mathbf{v})}{\mathbf{v}^t S_W \mathbf{v}} = 0$$

$$\Rightarrow S_B \mathbf{v} - \underbrace{\frac{\mathbf{v}^t S_B \mathbf{v}}{\mathbf{v}^t S_W \mathbf{v}}}_{=\lambda}(S_W \mathbf{v}) = 0$$

$$\Rightarrow \underbrace{S_B \mathbf{v} = \lambda S_W \mathbf{v}}_{\textit{generalized eigenvalue problem}}$$

$$\mathbf{S}_B \mathbf{v} = \lambda \mathbf{S}_W \mathbf{v}$$

- If $\mathbf{S}_W$ has full rank (the inverse exists), we can convert this to a standard eigenvalue problem

$$\mathbf{S}_W^{-1} \mathbf{S}_B \mathbf{v} = \lambda \mathbf{v}$$

- But $\mathbf{S}_B\mathbf{x}$ for any vector $\mathbf{x}$, points in the same direction as $\boldsymbol{\mu}_1$ - $\boldsymbol{\mu}_2$

$$\mathbf{S}_B \mathbf{x} = (\mu_1 - \mu_2)(\mu_1 - \mu_2)^t \mathbf{x} = (\mu_1 - \mu_2)\overbrace{\left((\mu_1 - \mu_2)^t \mathbf{x}\right)}^{\alpha} = \alpha(\mu_1 - \mu_2)$$

- Based on this, we can solve the eigenvalue problem directly

$$\boxed{\mathbf{v} = \mathbf{S}_W^{-1}(\mu_1 - \mu_2)}$$

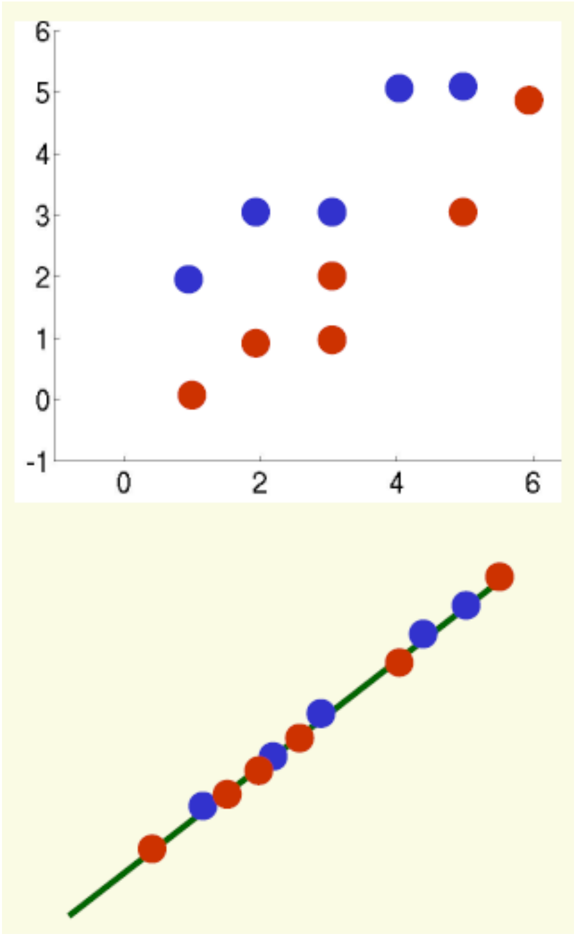
$$\mathbf{S}_W^{-1}\mathbf{S}_B\underbrace{\left[\mathbf{S}_W^{-1}(\mu_1 - \mu_2)\right]}_{\mathbf{v}} = \mathbf{S}_W^{-1}\left[\alpha(\mu_1 - \mu_2)\right] = \underbrace{\alpha}_{\lambda}\underbrace{\left[\mathbf{S}_W^{-1}(\mu_1 - \mu_2)\right]}_{\mathbf{v}}$$

Example

- Data
  - Class 1 has 5 samples
    c_1=[(1,2),(2,3),(3,3),(4,5),(5,5)]
  - Class 2 has 6 samples
    c_2=[(1,0),(2,1),(3,1),(3,2),(5,3),(6,5)]
- Arrange data in 2 separate matrices

$$c_1 = \begin{bmatrix} 1 & 2 \\ \vdots & \vdots \\ 5 & 5 \end{bmatrix} \qquad c_2 = \begin{bmatrix} 1 & 0 \\ \vdots & \vdots \\ 6 & 5 \end{bmatrix}$$

- Notice that PCA performs very poorly on this data because the direction of largest variance is not helpful for classification



- First compute the mean for each class

$$\mu_1 = mean(c_1) = [3 \quad 3.6]^t \qquad \mu_2 = mean(c_2) = [3.3 \quad 2]^t$$

- Compute scatter matrices S_1 and S_2 for each class

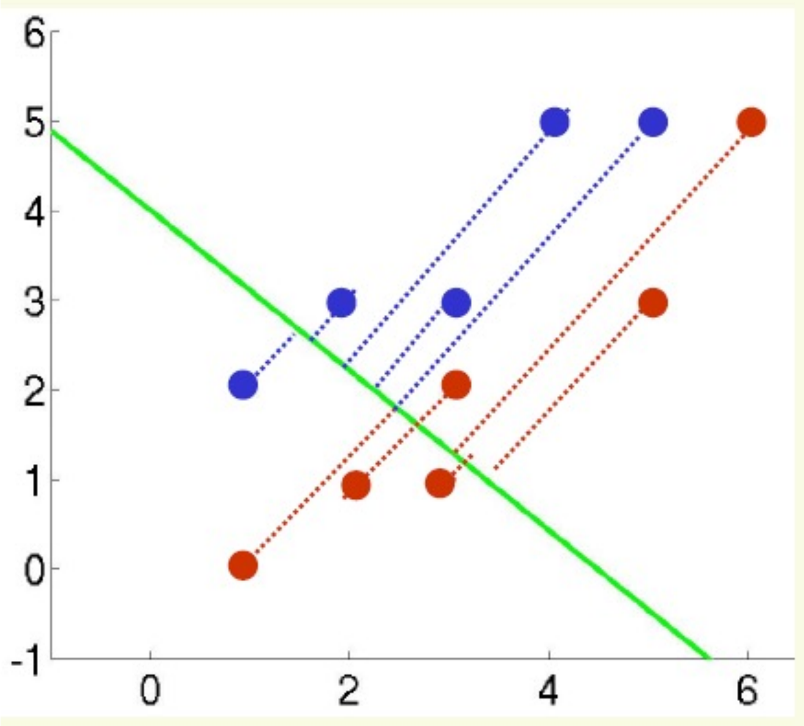
$$S_1 = 4 * cov(c_1) = \begin{bmatrix} 10 & 8.0 \\ 8.0 & 7.2 \end{bmatrix} \qquad S_2 = 5 * cov(c_2) = \begin{bmatrix} 17.3 & 16 \\ 16 & 16 \end{bmatrix}$$

- Within class scatter: $S_W = S_1 + S_2 = \begin{bmatrix} 27.3 & 24 \\ 24 & 23.2 \end{bmatrix}$
  - it has full rank, don't have to solve for eigenvalues
- The inverse of S_W is: $S_W^{-1} = inv(S_W) = \begin{bmatrix} 0.39 & -0.41 \\ -0.41 & 0.47 \end{bmatrix}$

- Finally, the optimal line direction **v** is:

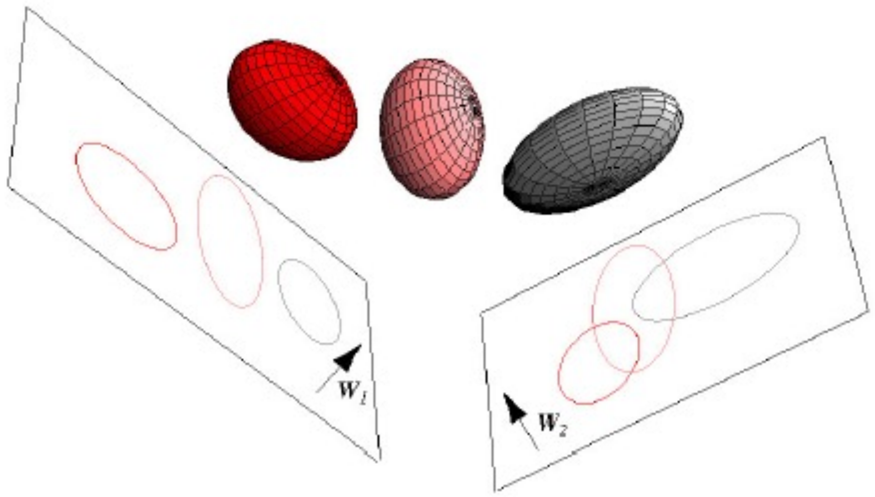
$$v = S_W^{-1}(\mu_1 - \mu_2) = \begin{bmatrix} -0.79 \\ 0.89 \end{bmatrix}$$

- As long as the line has the right direction, its exact position does not matter
- The last step is to compute the actual 1D vector ***y***
  - Separately for each class



Multiple Discriminant Analysis

- Can generalize FLD to multiple classes
  - In case of ***c*** classes, we can reduce dimensionality to 1, 2, 3,…, **c-1** dimensions
  - Project sample $\mathbf{x_i}$ to a linear subspace $\mathbf{y_i = V^t x_i}$
  - ***V*** is called projection matrix



- Within class scatter matrix:

$$S_W = \sum_{i=1}^{c} S_i = \sum_{i=1}^{c} \sum_{x_k \in \, class \; i} (x_k - \mu_i)(x_k - \mu_i)^t$$

- Between class scatter matrix

$$S_B = \sum_{i=1}^{c} n_i (\mu_i - \mu)(\mu_i - \mu)^t$$

maximum rank is c -1

mean of all data

mean of class i

- Objective function

$$J(V) = \frac{\det\left(V^t S_B V\right)}{\det\left(V^t S_W V\right)}$$

$$J(V) = \frac{\det\left(V^t S_B V\right)}{\det\left(V^t S_W V\right)}$$

- Solve generalized eigenvalue problem

$$S_B v = \lambda S_W v$$

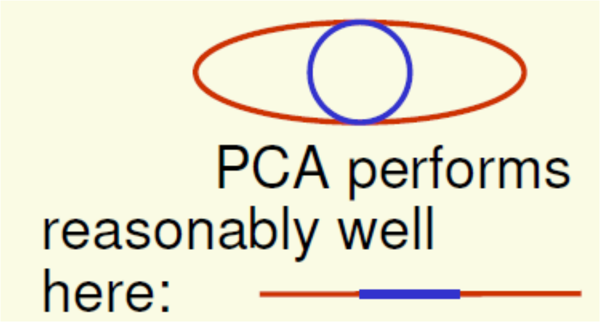
- There are at most **c-1** distinct eigenvalues
  - with $\mathbf{v_1 \ldots v_{c-1}}$ corresponding eigenvectors
- The optimal projection matrix **V** to a subspace of dimension **k** is given by the eigenvectors corresponding to the largest **k** eigenvalues
- Thus, we can project to a subspace of dimension at most **c-1**

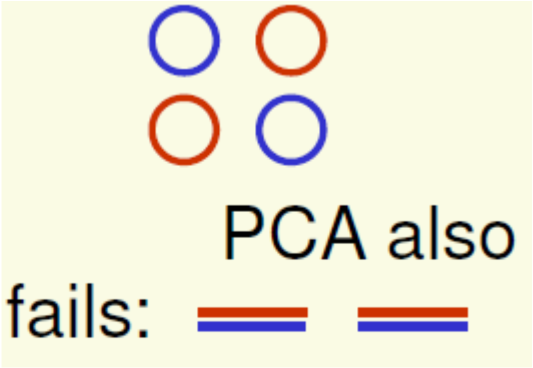
FDA and MDA Drawbacks

- Reduces dimension only to **k = c-1**
  - Unlike PCA where dimension can be chosen to be smaller or larger than **c-1**
- For complex data, projection to even the best line may result in non-separable projected samples

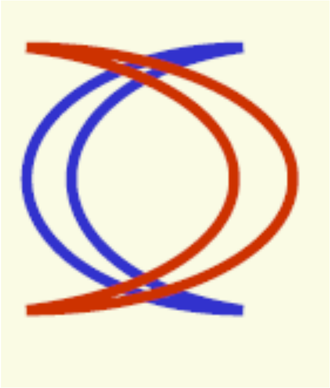
FDA and MDA Drawbacks

- FDA/MDA will fail:
  - If **J(v)** is always 0: when $\boldsymbol{\mu_1=\mu_2}$





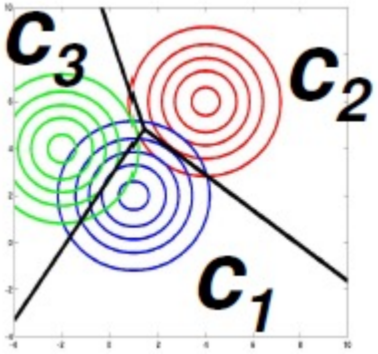
- If **J(v)** is always small: classes have large overlap when projected to any line (PCA will also fail)



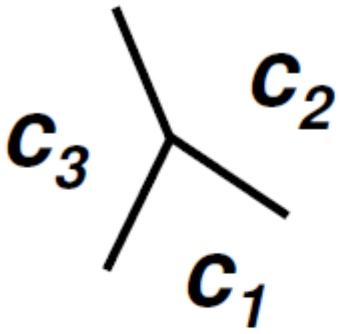
Generative vs. Discriminative Approaches

Parametric Methods vs. Discriminant Functions

- Assume the shape of density for classes is known $\mathbf{p_1(x|\theta_1), p_2(x|\theta_2),\dots}$
- Estimate $\boldsymbol{\theta}_1$, $\boldsymbol{\theta}_2$,… from data
- Use a Bayesian classifier to find decision regions



- Assume discriminant functions are of known shape $\mathbf{l(\theta_1), l(\theta_2)}$, with parameters $\boldsymbol{\theta}_1$, $\boldsymbol{\theta}_2$,…
- Estimate $\boldsymbol{\theta}_1$, $\boldsymbol{\theta}_2$,… from data
- Use discriminant functions for classification



Parametric Methods vs. Discriminant Functions

- In theory, Bayesian classifier minimizes the risk
  - In practice, we may be uncertain about ***our assumptions*** about the models
  - In practice, we may ***not really need*** the actual density functions
- Estimating accurate density functions is much harder than estimating accurate discriminant functions
  - Why solve a harder problem than needed?

Generative vs. Discriminative Models

Training classifiers involves estimating f: X → Y, or P(Y|X)

Discriminative classifiers

1. Assume some functional form for P(Y|X)
2. Estimate parameters of P(Y|X) directly from training data

Generative classifiers

1. Assume some functional form for P(X|Y), P(X)
2. Estimate parameters of P(X|Y), P(X) directly from training data
3. Use Bayes rule to calculate $P(Y|X= x_i)$

Generative vs. Discriminative Example

- The task is to determine the language that someone is speaking
- Generative approach:
  - Learn each language and determine which language the speech belongs to
- Discriminative approach:
  - Determine the linguistic differences without learning any language – a much easier task!

Generative vs. Discriminative Taxonomy

- Generative Methods
  - Model class-conditional pdfs and prior probabilities
  - "Generative" since sampling can generate synthetic data points
  - Popular models
    - Multi-variate Gaussians, Naïve Bayes
    - Mixtures of Gaussians, Mixtures of experts, Hidden Markov Models (HMM)
    - Sigmoidal belief networks, Bayesian networks, Markov random fields

- Discriminative Methods
  - Directly estimate posterior probabilities
  - No attempt to model underlying probability distributions
  - Focus computational resources on given task– better performance
  - Popular models
    - Logistic regression
    - SVMs
    - Traditional neural networks
    - Nearest neighbor
    - Conditional Random Fields (CRF)

Generative Approach

- Advantage
  - ***Prior information*** about the structure of the data is often most naturally specified through a generative model P(X|Y)
    - For example, for male faces, we would expect to see heavier eyebrows, a more square jaw, etc.
- Disadvantages
  - The generative approach does not directly target the classification model P(Y|X) since the goal of generative training is P(X|Y)
  - If the data x are complex, finding a suitable generative data model P(X|Y) is a difficult task
  - Since each generative model is separately trained for each class, there is ***no competition*** amongst the models to explain the data
  - The decision boundary between the classes may have a simple form, even if the data distribution of each class is complex

Discriminative Approach

- Advantages
  - The discriminative approach directly addresses finding an accurate classifier P(Y|X) based on modelling the decision boundary, as opposed to the class conditional data distribution
  - Whilst the data from each class may be distributed in a complex way, it could be that the decision boundary between them is relatively easy to model
- Disadvantages
  - Discriminative approaches are usually trained as “black-box” classifiers, with ***<u>little prior knowledge</u>*** built used to describe how data for a given class is distributed
  - ***<u>Domain knowledge</u>*** is often more easily expressed using the generative framework

Linear Discriminant Functions

LDF: Introduction

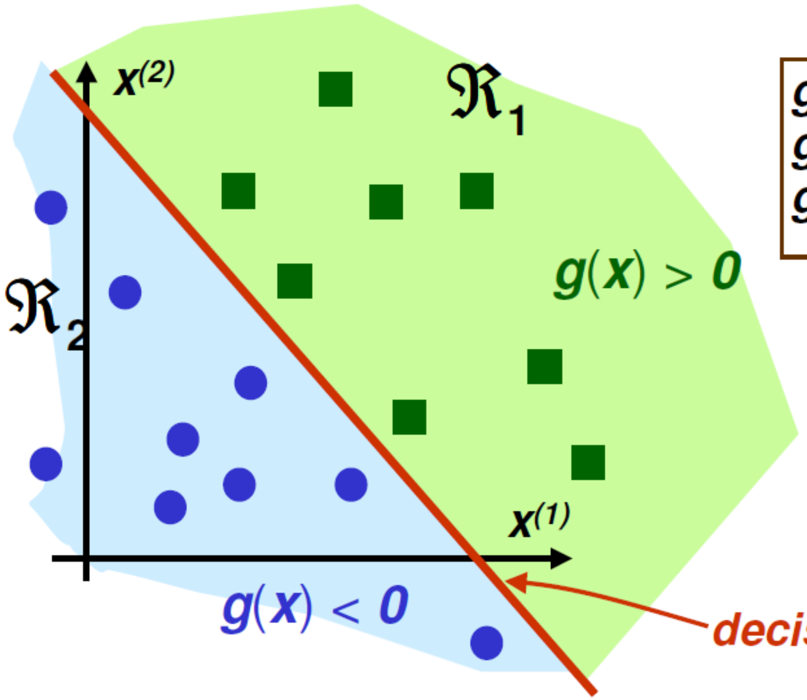
- Discriminant functions can be more general than linear
- For now, focus on linear discriminant functions
  - Simple model (should try simpler models first)
  - Analytically tractable
- Linear Discriminant functions are optimal for ***Gaussian distributions*** with ***equal covariance***
- May not be optimal for other data distributions, but they are very simple to use

LDF: Two Classes

- A discriminant function is linear if it can be written as

$$g(x) = w^t x + w_0$$

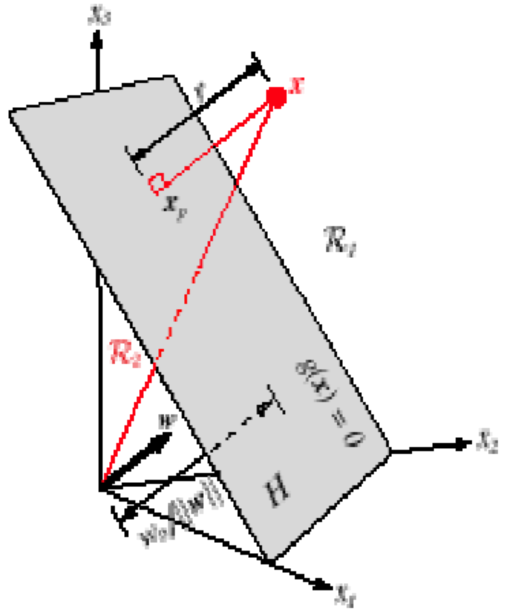
- w is called the weight vector and w_0 is called the bias or threshold



$$\begin{array}{ll} g(x) > 0 & \Rightarrow x \in \text{class } 1 \\ g(x) < 0 & \Rightarrow x \in \text{class } 2 \\ g(x) = 0 & \Rightarrow \text{either class} \end{array}$$

LDF: Two Classes

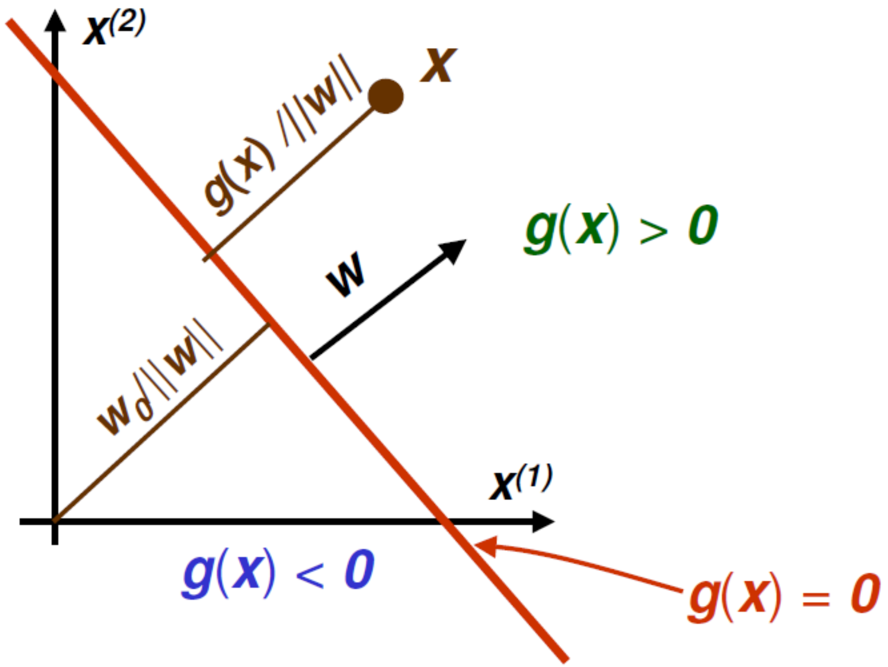
- Decision boundary $\mathbf{g(x) = w^t x + w_0 = 0}$ is a hyperplane
  - Set of vectors $\mathbf{x}$, which for some scalars $a_0, \dots, a_d$, satisfy $a_0 + a_1 x^{(1)} + \dots + a_d x^{(d)} = 0$
  - A hyperplane is:
  - a point in 1D
  - a line in 2D
  - a plane in 3D



LDF: Two Classes

$$g(x) = w^t x + w_0$$

- **w** determines the orientation of the decision hyperplane
- $\mathbf{w_0}$ determines the location of the decision surface



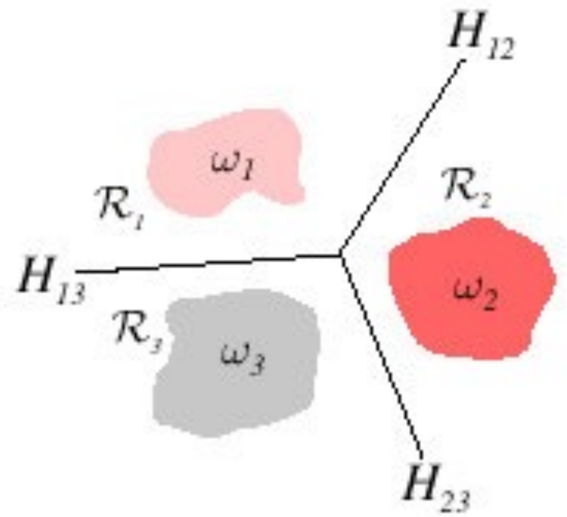
LDF: Multiple Classes

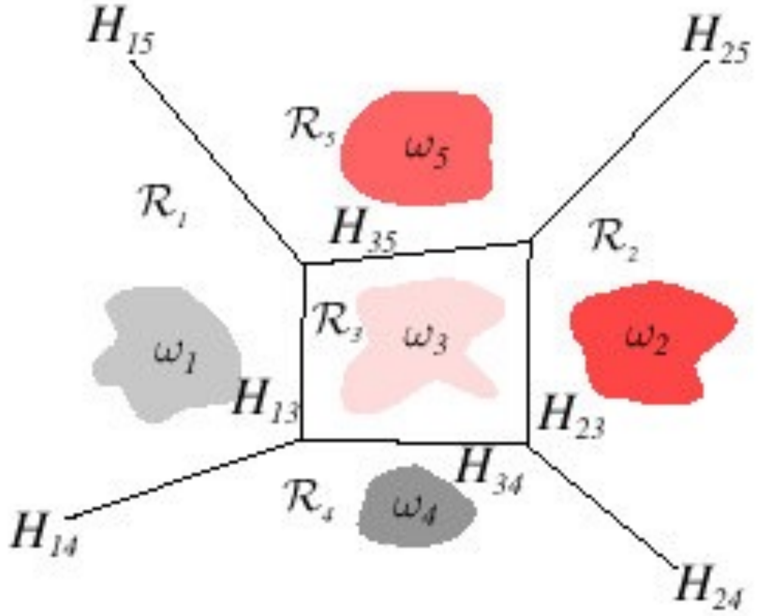
- Suppose we have **m** classes
- Define **m** linear discriminant functions

$$\mathbf{g_i (x) = w_i^t x + w_{i0}}$$

- Given **x**, assign to class $\mathbf{c_i}$ if
  - $\mathbf{g_i (x) > g_j(x), i \neq j}$
- Such a classifier is called a linear machine
- A linear machine divides the feature space into **c** decision regions, with $\mathbf{g_i(x)}$ being the largest discriminant if **x** is in the region R_i

LDF: Multiple Classes





LDF: Multiple Classes

- For two contiguous regions $\mathbf{R_i}$ and $\mathbf{R_j}$, the boundary that separates them is a portion of the hyperplane $\mathbf{H_{ij}}$ defined by:

$$g_i(x) = g_j(x) \iff w_i^t x + w_{i0} = w_j^t x + w_{j0}$$
$$\iff (w_i - w_j)^t x + (w_{i0} - w_{j0}) = 0$$

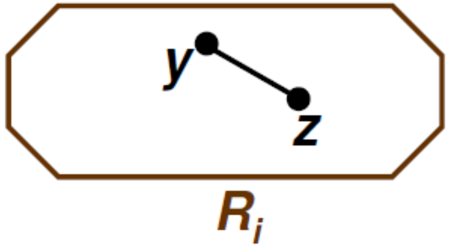
- Thus $\mathbf{w_i - w_j}$ is normal to $\mathbf{H_{ij}}$
- The distance from $\mathbf{x}$ to $\mathbf{H_{ij}}$ is given by:

$$d(x, H_{ij}) = \frac{g_i(x) - g_j(x)}{\|w_i - w_j\|}$$

LDF: Multiple Classes

- Decision regions for a linear machine are convex

$$y, z \in R_i \Rightarrow \alpha y + (1-\alpha) z \in R_i$$



$$\forall j \neq i \quad g_i(y) \geq g_j(y) \text{ and } g_i(z) \geq g_j(z) \Leftrightarrow$$
$$\Leftrightarrow \forall j \neq i \quad g_i(\alpha y + (1-\alpha) z) \geq g_j(\alpha y + (1-\alpha) z)$$

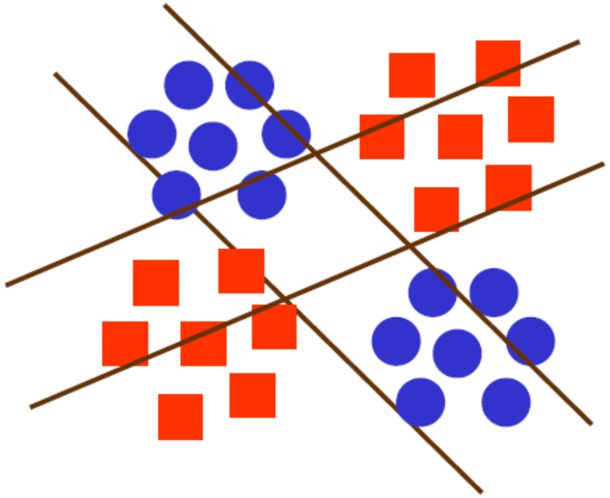
- In particular, decision regions must be spatially contiguous

R_j is a valid decision region

R_j is not a valid decision region

LDF: Multiple Classes

- Thus applicability of linear machine mostly limited to unimodal conditional densities **p(x|θ)**
  - Example:



- Need non-contiguous decision regions
- Linear machine will fail