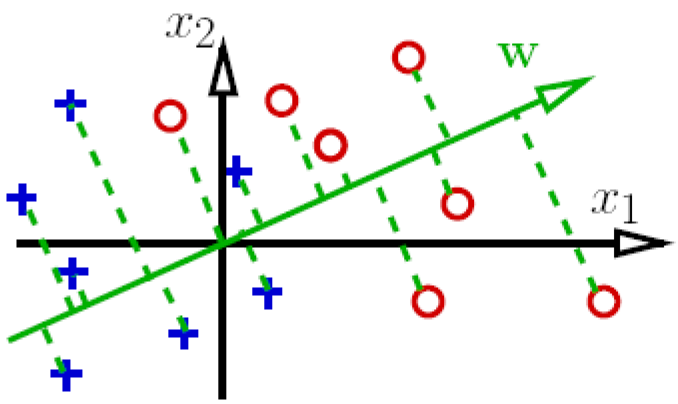
Classification

Discriminant Analysis

slides thanks to Greg Shakhnarovich (CS195-5, Brown Univ., 2006)

Distribution in 1D projection

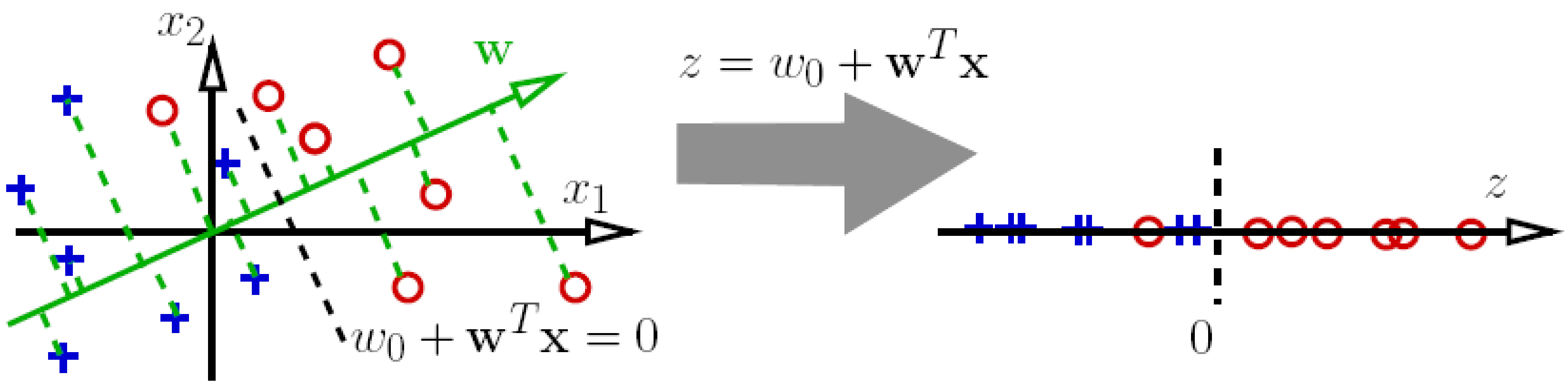


- Consider a scalar *projection*

$$f : \mathbf{x} \;\rightarrow\; w_0 + \mathbf{w}^T \mathbf{x}$$

- We can study how well the projected values corresponding to different classes are separated
  - This is a function of $\mathbf{w}$; some projections may be better than others.

Distribution in 1D projection



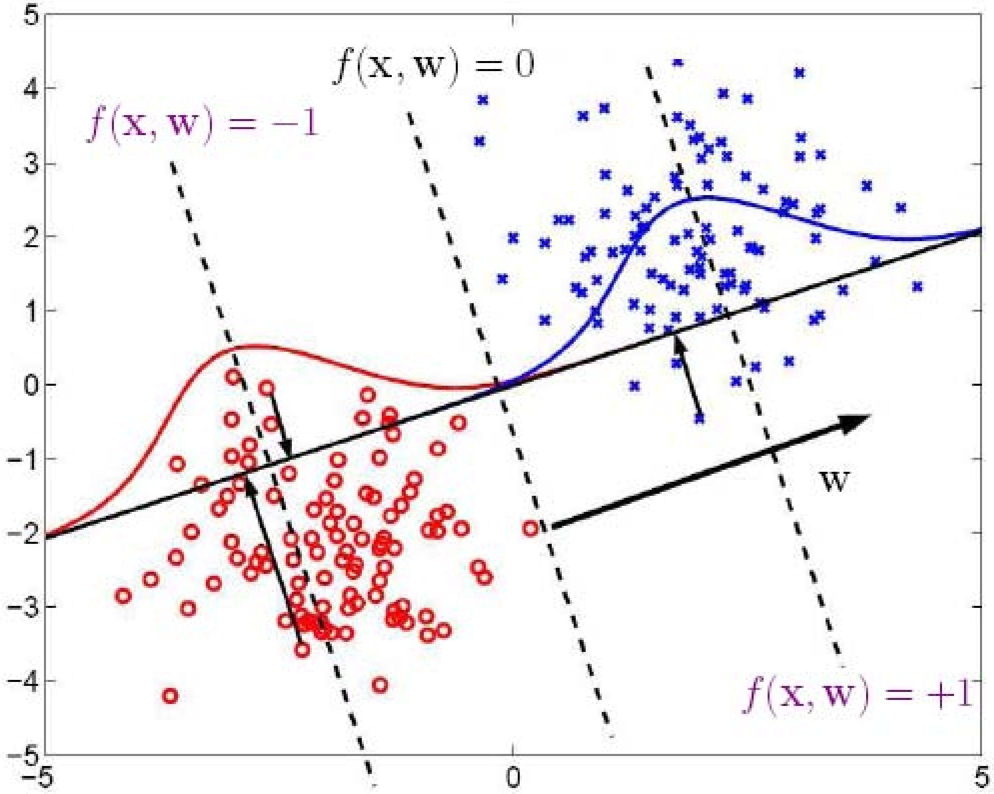
- Consider a scalar *projection*

$$f : \mathbf{x} \;\rightarrow\; w_0 + \mathbf{w}^T\mathbf{x}$$

- We can study how well the projected values corresponding to different classes are separated
  - This is a function of $\mathbf{w}$; some projections may be better than others.

Linear discriminant and dimensionality reduction

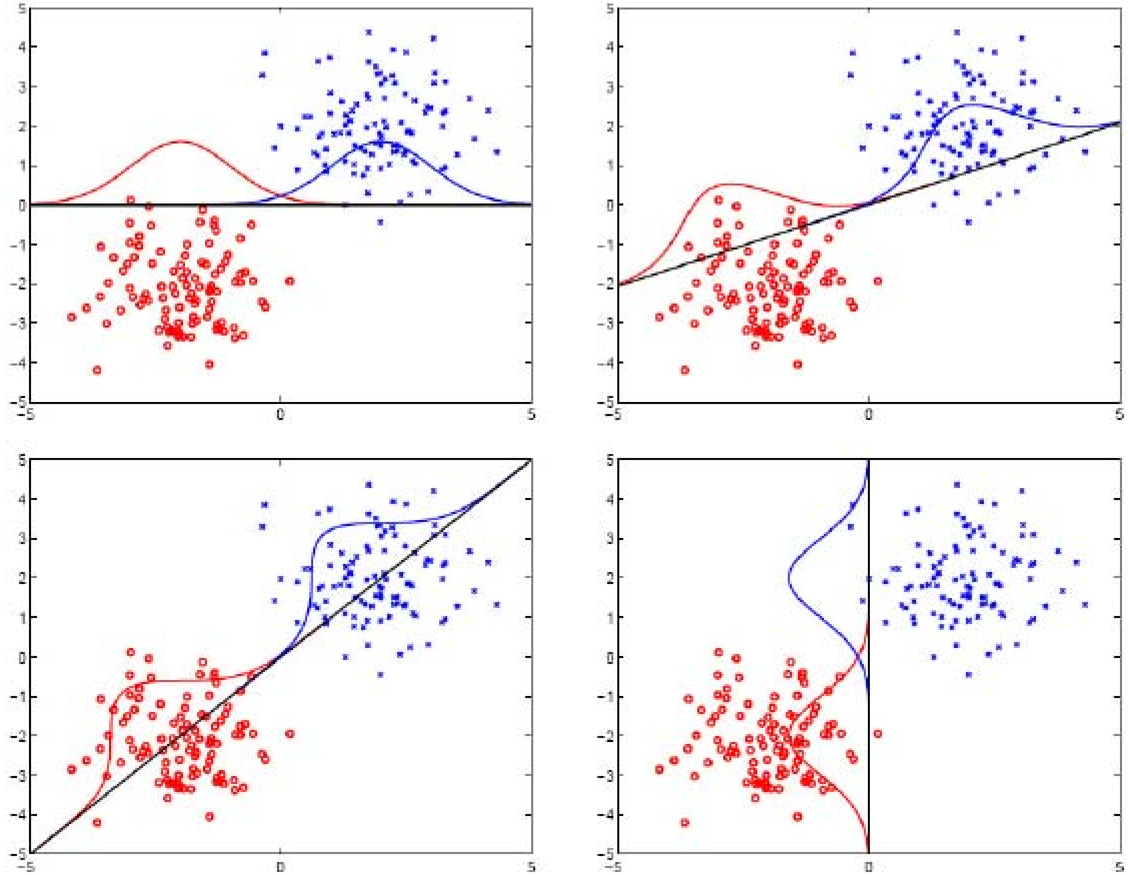
The *discriminant function* $f(\mathbf{x};\mathbf{w}) = w_0 + \mathbf{w}^T\mathbf{x}$ reduces the dimension of examples from d to 1; the components orthogonal to $\mathbf{w}$ become irrelevant.



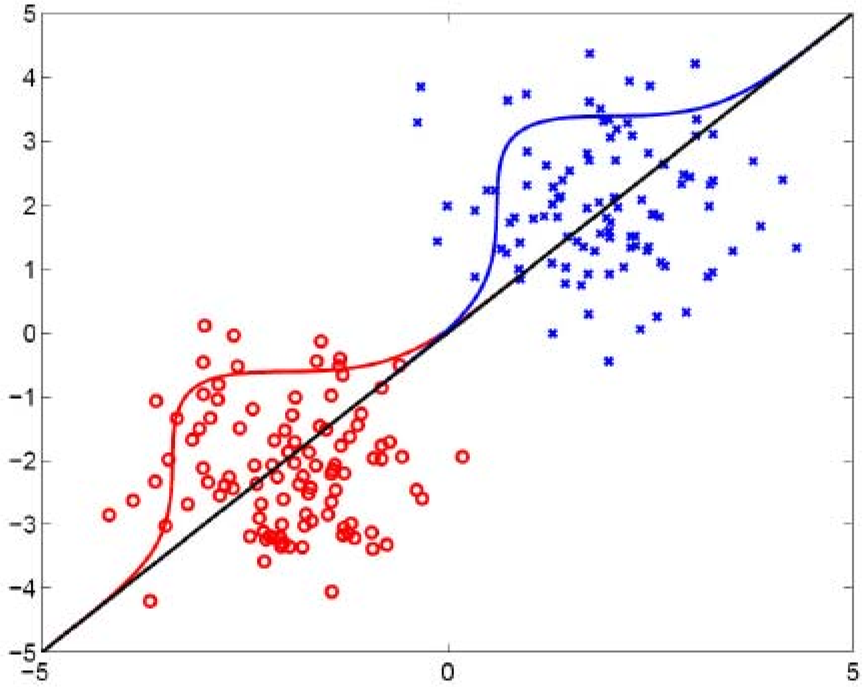
$$\hat{y} = +1 \Leftrightarrow f(\mathbf{x};\mathbf{w}) > 0$$

Projections and classification

What objecive are we optimizing the 1D projection for?



Objective: class separation



- We want to minimize “overlap” between projections of the two classes.
- One approach: make the class projections a) compact, b) far apart.
- An obvious idea: maximize separation between the projected means.

Separation of the means

- N_{+1} examples of class $+1$, N_{-1} examples of class -1.
- The *empirical mean* of each class:

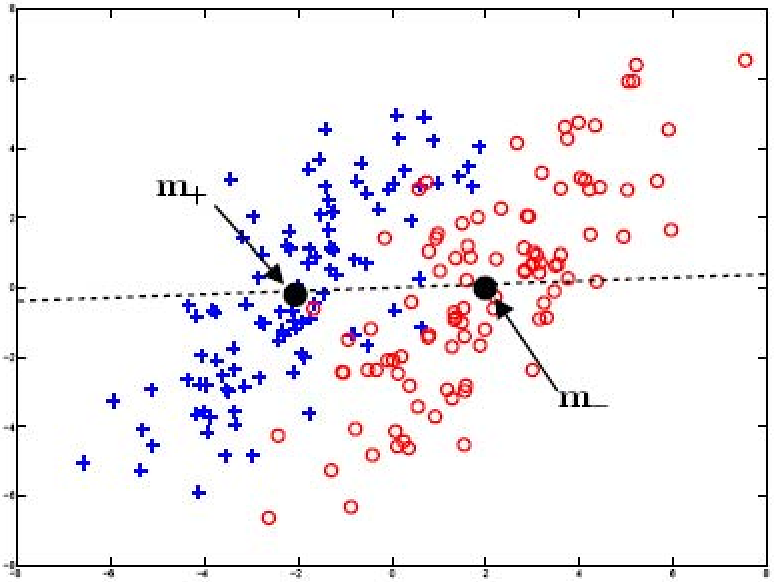
$$\mathbf{m}_{+1} = \frac{1}{N_{+1}} \sum_{y_i=+1} \mathbf{x}_i, \qquad \mathbf{m}_{-1} = \frac{1}{N_{-1}} \sum_{y_i=-1} \mathbf{x}_i$$

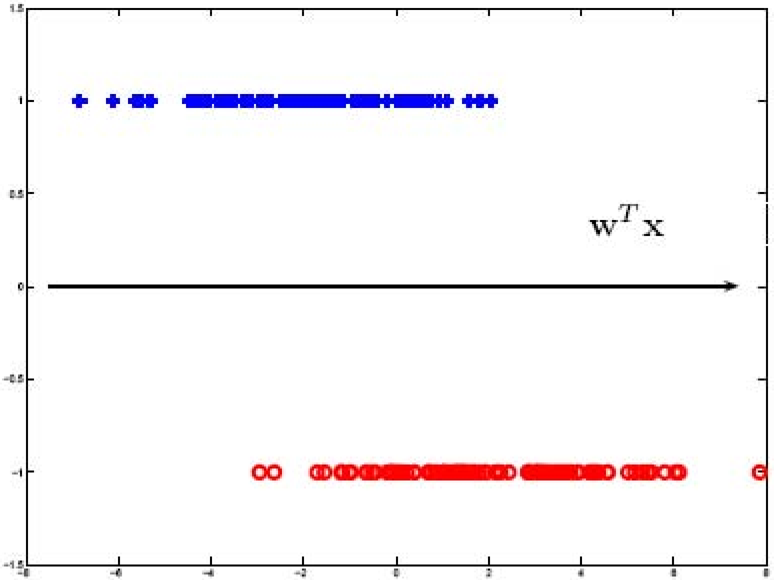
- We can look for projection $\hat{\mathbf{w}}$ such that

$$\hat{\mathbf{w}} \;=\; \operatorname*{argmax}_{\mathbf{w}} \mathbf{w}^T(\mathbf{m}_{+1} - \mathbf{m}_{-1})$$

Separation of the means: example

$$\hat{\mathbf{w}} = \underset{\|\mathbf{w}\|=1}{\operatorname{argmax}} \mathbf{w}^T(\mathbf{m}_{+1} - \mathbf{m}_{-1})$$





- Also want to make projection of each class "compact"...

Fisher's linear discriminant analysis

- Criterion to be maximized:

$$J_{Fisher}(\mathbf{w}) = \frac{\text{separation between projected means}^2}{\text{sum of projected within-class variances}}$$

- Numerator: *between-class scatter* $\left(\mathbf{w}^T(\mathbf{m}_{+1} - \mathbf{m}_{-1})\right)^2$
- Denominator: *within-class scatter* $\mathbf{w}^T\left(N_{-1}\mathbf{S}_{-1} + N_{+1}\mathbf{S}_{+1}\right)\mathbf{w}$, where

$$\mathbf{S}_c = \frac{1}{N_c}\sum_{y_i=c}(\mathbf{x}_i - \mathbf{m}_c)(\mathbf{x}_i - \mathbf{m}_c)^T.$$

  - The denominator is the sum of estimated 1D class covariances, after data are projected to $\mathbf{w}$, weighted by number of samples in each class.

Fisher's LDA

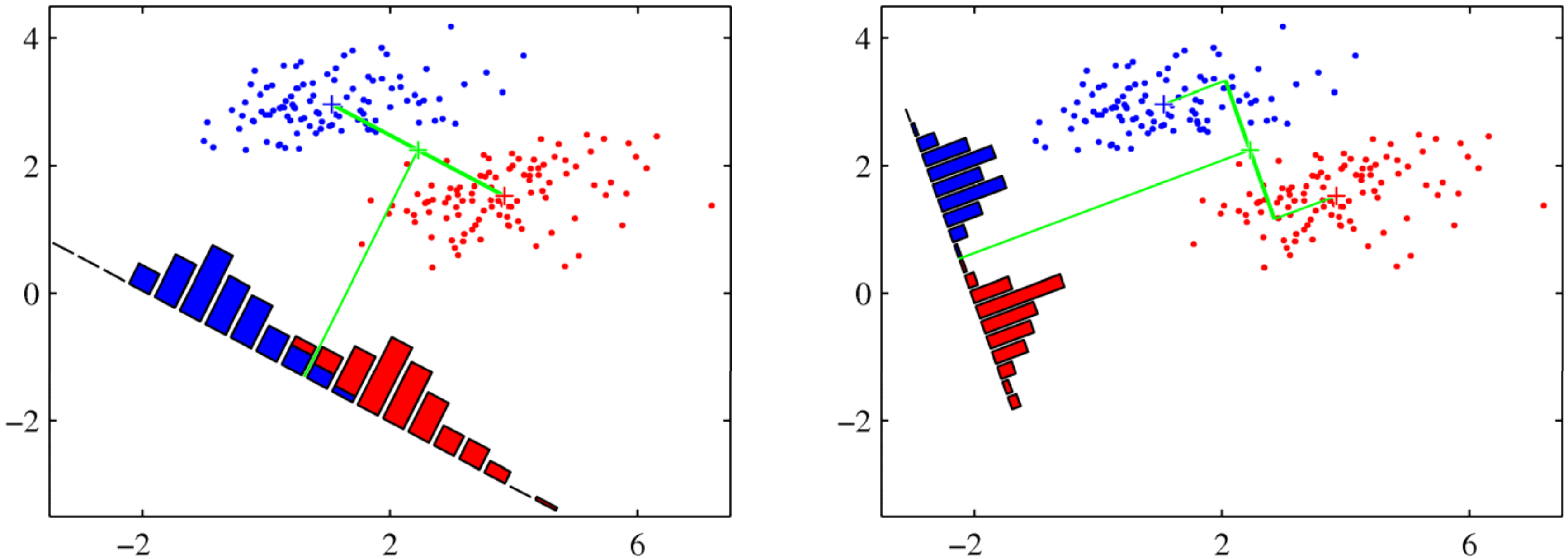
$$J_{Fisher}(\mathbf{w}) = \frac{\left(\mathbf{w}^T(\mathbf{m}_{+1} - \mathbf{m}_{-1})\right)^2}{\mathbf{w}^T\left(N_{-1}\mathbf{S}_{-1} + N_{+1}\mathbf{S}_{+1}\right)\mathbf{w}}$$

- Best 1D projection: $\hat{\mathbf{w}} = \mathrm{argmax}_{\mathbf{w}}\, J_{Fisher}(\mathbf{w})$
- Setting the derivative of J w.r.t. $\mathbf{w}$ to zero, get solution:

$$\hat{\mathbf{w}} \propto \left(N_{-1}\mathbf{S}_{-1} + N_{+1}\mathbf{S}_{+1}\right)^{-1}(\mathbf{m}_{+1} - \mathbf{m}_{-1})$$

Notation: $\propto$ means "proportional to", up to a constant factor.

Example of applying Fisher's LDA



maximize separation of means

maximize Fisher's LDA criterion $\rightarrow$ better class separation

Using LDA for classification in one dimension

- Fisher's LDA gives an optimal choice of **w**, the vector for projection down to one dimension.
- For classification, we still need to select a threshold to compare projected values to. Two possibilities:
  - No explicit probabilistic assumptions. Find threshold which minimizes empirical classification error.
  - Make assumptions about data distributions of the classes, and derive theoretically optimal decision boundary.
    - Usual choice for class distributions is multivariate Gaussian.
    - We also will need a bit of decision theory.

Decision theory

To minimize classification error:

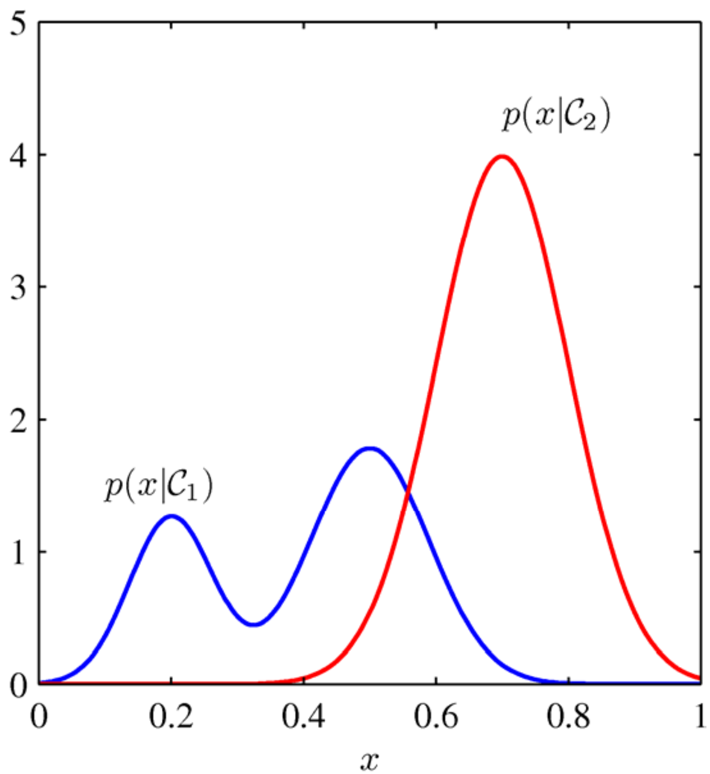
$$\hat{y} = \arg\max_{C} p(C \mid \mathbf{x}) \qquad \approx$$

At a given point **x** in feature space, choose as the predicted class the class that has the greatest probability at **x**.

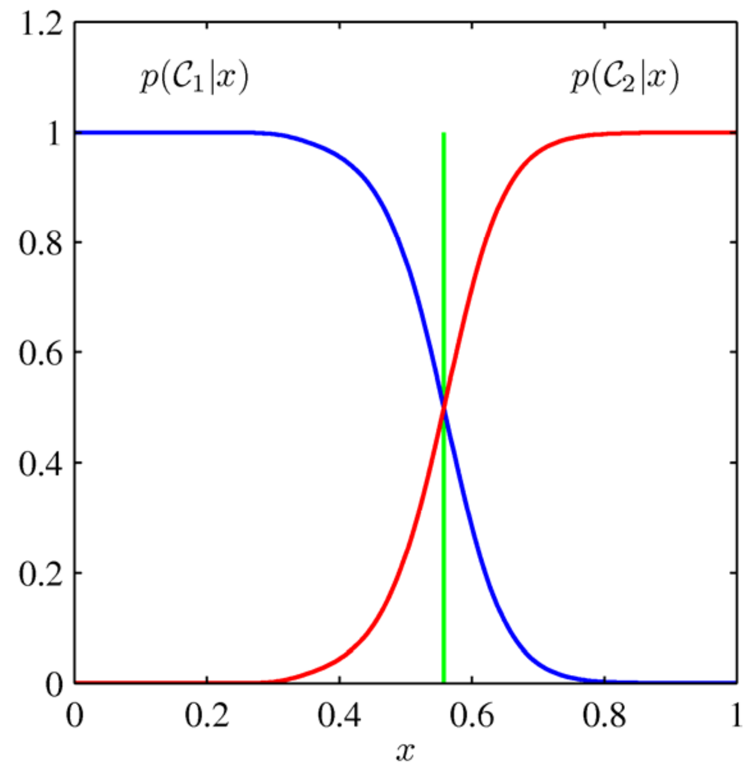
Decision theory

$$\hat{y} = \arg\max_{C} p(C \mid \mathbf{x}) \qquad \approx$$

At a given point **x** in feature space, choose as the predicted class the class that has the greatest probability at **x**.



probability densities for classes C_1 and C_2



relative probabilities for classes C_1 and C_2

MATLAB interlude

Classification via discriminant analysis,
using the classify() function.

Data for each class modeled as multivariate Gaussian.

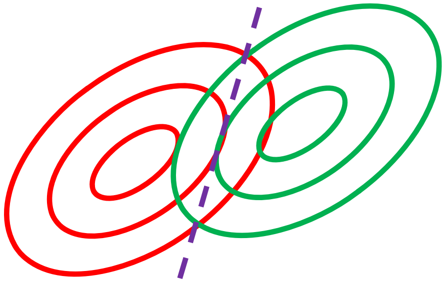
matlab_demo_06.m

```
class = classify( sample, training, group, 'type' )
```

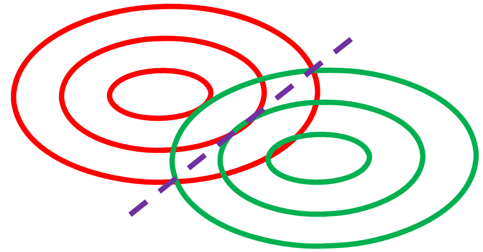
- class: predicted test labels
- sample: test data
- training: training data
- group: training labels
- 'type': model for class covariances

MATLAB classify() function

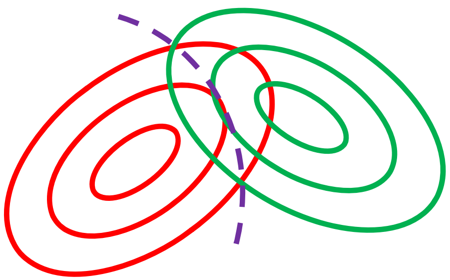
Models for class covariances



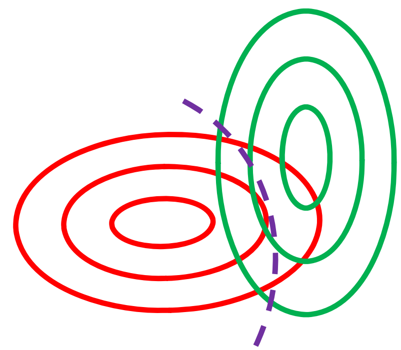
`'linear'`:
all classes have same covariance matrix
→ linear decision boundary



`'diaglinear'`:
all classes have same diagonal covariance matrix
→ linear decision boundary



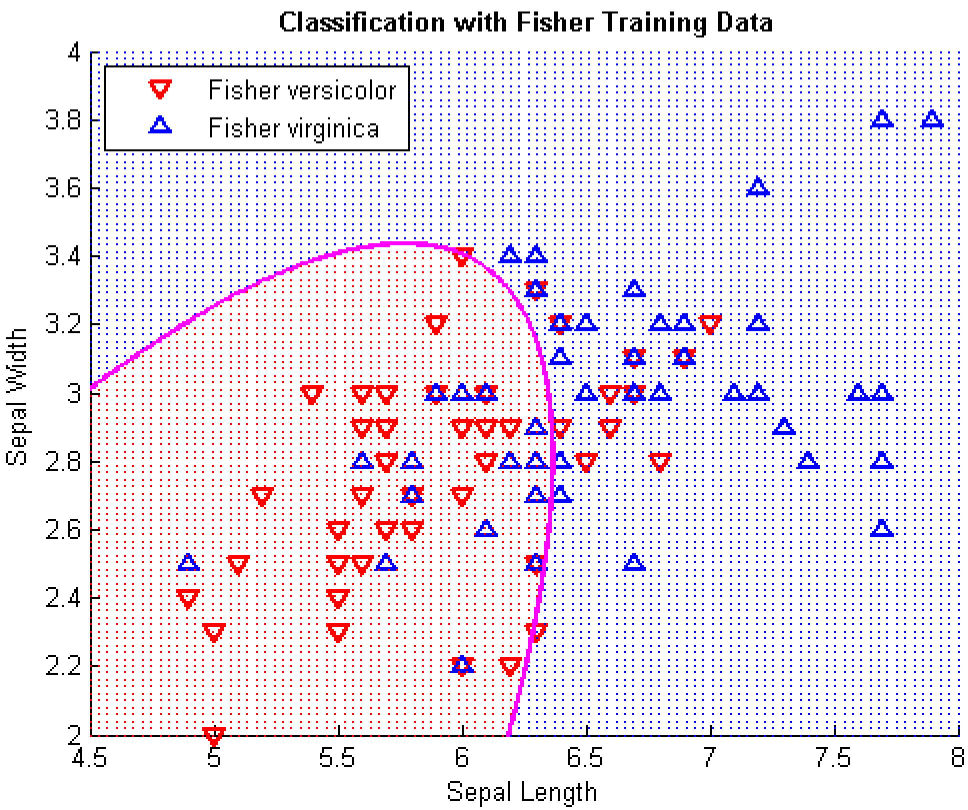
`'quadratic'`:
classes have different covariance matrices
→ quadratic decision boundary



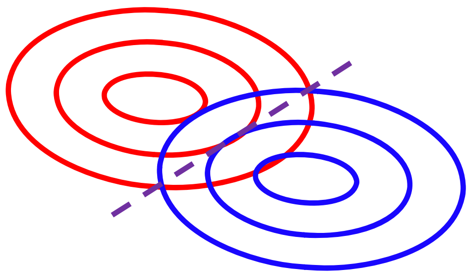
`'diagquadratic'`:
classes have different diagonal covariance matrices
→ quadratic decision boundary

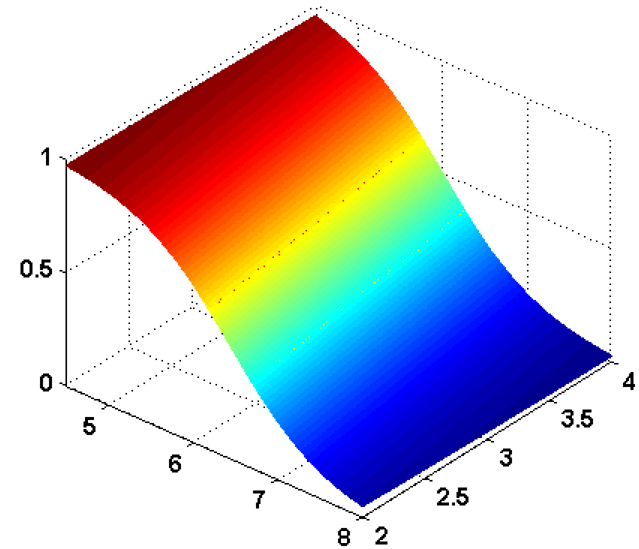
MATLAB classify() function

Example with `'quadratic'` model of class covariances



Relative class probabilities for LDA





`'linear'`:
all classes have same covariance matrix
→ linear decision boundary

relative class probabilities have exactly same sigmoidal form as in logistic regression