

# CS195-5 : Introduction to Machine Learning

## Lecture 7

Greg Shakhnarovich

September 20 2006

Revised October 24th, 2006

---

# Announcements

- PS1 clarifications:
  - P.5: “quadratic regression” means quadratic in  $\mathbf{x}$ .
  - P.5: Why not try higher order models?
- MLRG today: introduction to sampling methods

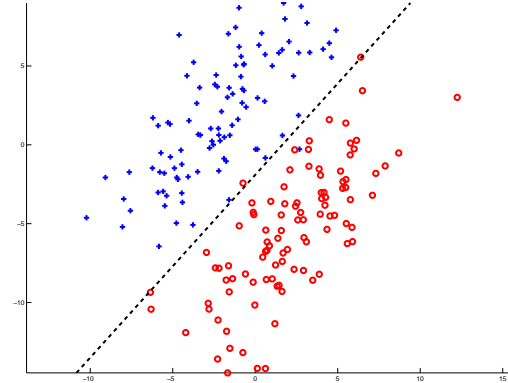
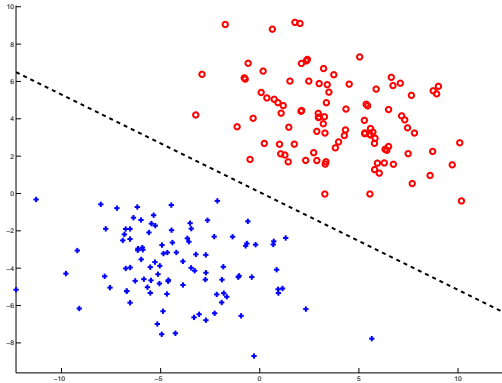
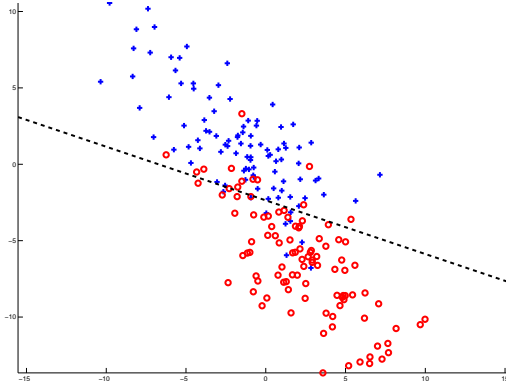
# Review

- Fisher's criterion:  $J_{Fisher}(\mathbf{w}) = \frac{\text{separation between projected means}^2}{\text{sum of projected within-class variances}}$ 
  - Resulting 1D projection:

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

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

- Decision boundary set by  $\hat{\mathbf{w}}^T \mathbf{x} + w_0 = 0$ .



---

## Linear separation of classes

- Classifying using a linear decision boundary (as in Fisher's method) effectively reduces the data dimension to 1.
- Important questions:
  - What's the optimal projection?
  - How does one set the *bias*  $w_0$ ?
  - Can we do better with more complex decision boundaries?

---

## Risk of a classifier

- The risk (expected loss) of a  $C$ -way classifier  $h(\mathbf{x})$ :

$$\begin{aligned} R(h) &= \int_{\mathbf{x}} \sum_{c=1}^C L(h(\mathbf{x}), c) p(\mathbf{x}, y = c) d\mathbf{x} \\ &= \int_{\mathbf{x}} \left[ \sum_{c=1}^C L(h(\mathbf{x}), c) p(y = c | \mathbf{x}) \right] p(\mathbf{x}) d\mathbf{x} \end{aligned}$$

- Clearly, it's enough to minimize the *conditional risk* for any  $\mathbf{x}$ :

$$R(h | \mathbf{x}) = \sum_{c=1}^C L(h(\mathbf{x}), c) p(y = c | \mathbf{x}).$$

---

## Conditional risk of a classifier

$$R(h | \mathbf{x}) = \sum_{c=1}^C L(h(\mathbf{x}), c) p(y = c | \mathbf{x})$$

---

## Conditional risk of a classifier

$$\begin{aligned} R(h | \mathbf{x}) &= \sum_{c=1}^C L(h(\mathbf{x}), c) p(y = c | \mathbf{x}) \\ &= 0 \cdot p(y = h(\mathbf{x}) | \mathbf{x}) + 1 \cdot \sum_{c \neq h(\mathbf{x})} p(y = c | \mathbf{x}) \end{aligned}$$

---

## Conditional risk of a classifier

$$\begin{aligned} R(h | \mathbf{x}) &= \sum_{c=1}^C L(h(\mathbf{x}), c) p(y = c | \mathbf{x}) \\ &= 0 \cdot p(y = h(\mathbf{x}) | \mathbf{x}) + 1 \cdot \sum_{c \neq h(\mathbf{x})} p(y = c | \mathbf{x}) = \sum_{c \neq h(\mathbf{x})} p(y = c | \mathbf{x}) \end{aligned}$$

---

## Conditional risk of a classifier

$$\begin{aligned} R(h | \mathbf{x}) &= \sum_{c=1}^C L(h(\mathbf{x}), c) p(y = c | \mathbf{x}) \\ &= 0 \cdot p(y = h(\mathbf{x}) | \mathbf{x}) + 1 \cdot \sum_{c \neq h(\mathbf{x})} p(y = c | \mathbf{x}) = \sum_{c \neq h(\mathbf{x})} p(y = c | \mathbf{x}) \\ &= 1 - p(y = h(\mathbf{x}) | \mathbf{x}). \end{aligned}$$

- Thus, to minimize conditional risk given  $\mathbf{x}$ , the classifier must decide

$$h(\mathbf{x}) = \operatorname{argmax}_c p(y = c | \mathbf{x}).$$

- This is the *best possible* classifier in terms of generalization, i.e. expected misclassification rate on new examples.

---

# Bayes rule

- Some terminology:

*class-conditional density*       $p_c(\mathbf{x}) = p(\mathbf{x} | y = c)$   
(also called *likelihood*)

*prior probability*       $P_c = p(y = c)$

*posterior probability*       $p(y = c | \mathbf{x})$

*compound density/probability of data*       $p(\mathbf{x}) = \sum_c p(\mathbf{x}, y = c) = \sum_c p_c(\mathbf{x})P_c.$

- Usually we don't have direct access to  $p(y | \mathbf{x})$ . But suppose we know  $p(\mathbf{x} | y)$  and  $p(y)$ .
- Bayes rule: Using the product rule  $p(a, b) = p(a | b) p(b) = p(b | a) p(a)$ ,

$$p(y | \mathbf{x}) = \frac{p(\mathbf{x} | y) p(y)}{p(\mathbf{x})}.$$

---

# Bayes classifier

$$p(y | \mathbf{x}) = \frac{p(\mathbf{x} | y) p(y)}{p(\mathbf{x})}.$$

- The classifier that minimizes conditional risk for given  $p(\mathbf{x} | y), p(y)$  is called the *Bayes classifier*

$$\begin{aligned} h^*(\mathbf{x}) &= \operatorname{argmax}_c p(y = c | \mathbf{x}) \\ &= \operatorname{argmax}_c \frac{p(\mathbf{x} | y = c) p(y = c)}{p(\mathbf{x})} \end{aligned}$$

---

# Bayes classifier

$$p(y | \mathbf{x}) = \frac{p(\mathbf{x} | y) p(y)}{p(\mathbf{x})}.$$

- The classifier that minimizes conditional risk for given  $p(\mathbf{x} | y), p(y)$  is called the *Bayes classifier*

$$\begin{aligned} h^*(\mathbf{x}) &= \operatorname{argmax}_c p(y = c | \mathbf{x}) \\ &= \operatorname{argmax}_c \frac{p(\mathbf{x} | y = c) p(y = c)}{p(\mathbf{x})} \\ &= \operatorname{argmax}_c p(\mathbf{x} | y = c) p(y = c) \end{aligned}$$

(Data probability term  $p(\mathbf{x})$  is equal for all  $c$ .)

---

# Bayes classifier

$$p(y | \mathbf{x}) = \frac{p(\mathbf{x} | y) p(y)}{p(\mathbf{x})}.$$

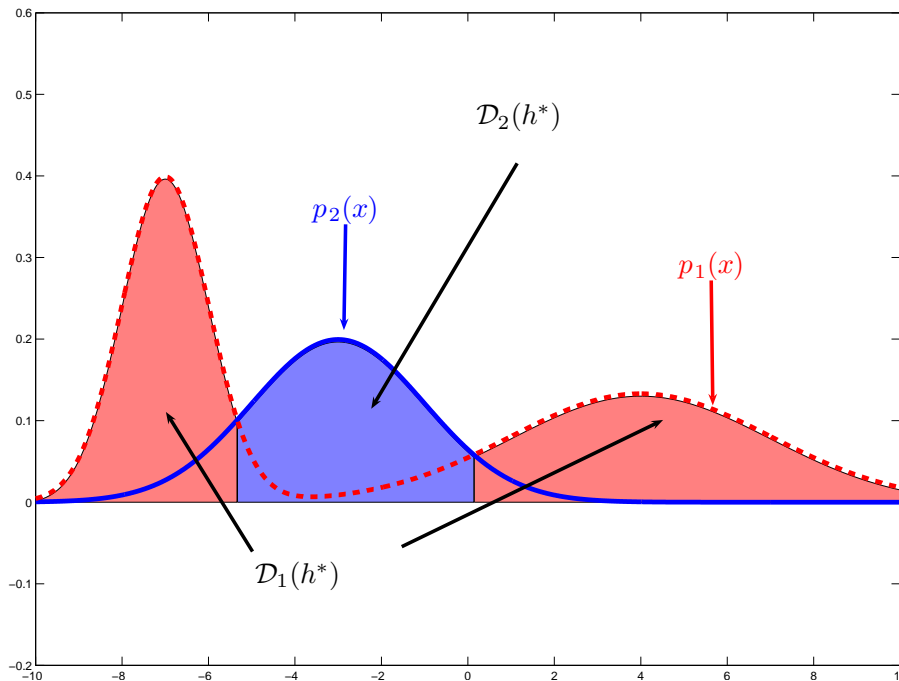
- The classifier that minimizes conditional risk for given  $p(\mathbf{x} | y), p(y)$  is called the *Bayes classifier*

$$\begin{aligned} h^*(\mathbf{x}) &= \operatorname{argmax}_c p(y = c | \mathbf{x}) \\ &= \operatorname{argmax}_c \frac{p(\mathbf{x} | y = c) p(y = c)}{p(\mathbf{x})} \\ &= \operatorname{argmax}_c p(\mathbf{x} | y = c) p(y = c) \\ &= \operatorname{argmax}_c \{ \log p_c(\mathbf{x}) + \log P_c \}. \end{aligned}$$

(Data probability term  $p(\mathbf{x})$  is equal for all  $c$ .)

# Optimal decision regions

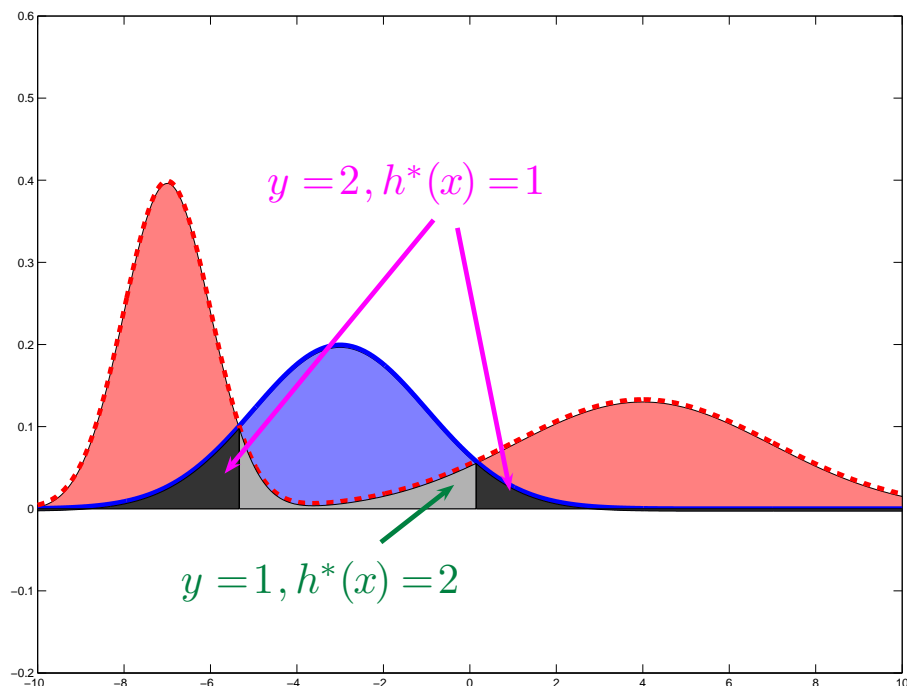
- A decision region is defined for each  $c$ :  $\mathcal{D}_c(h) = \{\mathbf{x} : h(\mathbf{x}) = c\}$ .



- If  $\forall c, P_c = 1/C$ , i.e. classes are equally likely, the optimal decision regions are simply

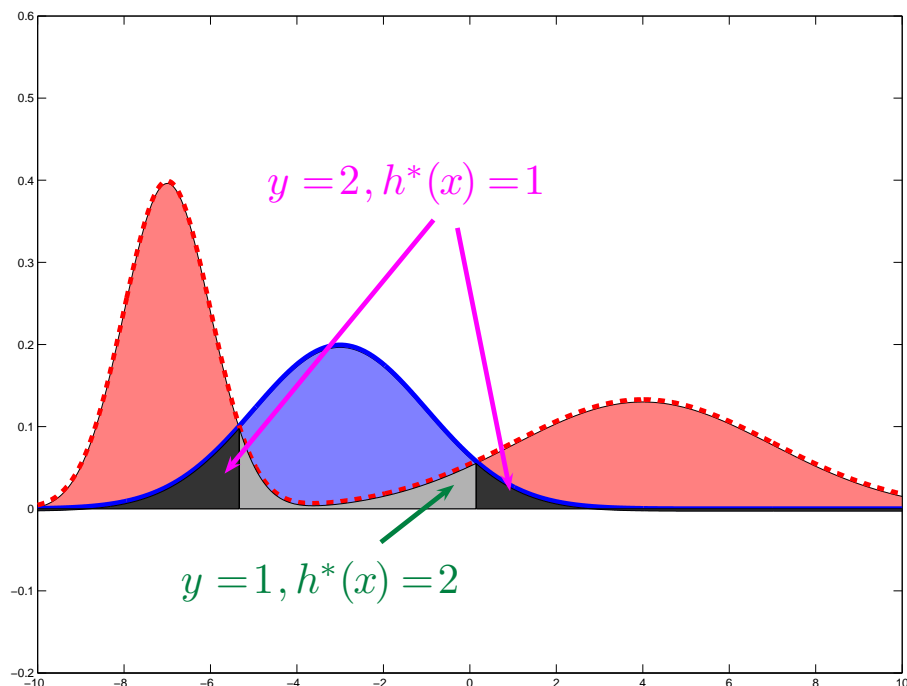
$$\mathcal{D}_c(h^*) = \{\mathbf{x} : c = \operatorname{argmax}_{c'} p_{c'}(\mathbf{x})\}.$$

# Bayes risk



- The risk (probability of error) of Bayes classifier  $h^*$  is called the *Bayes risk*  $R^*$ .
- This is the *minimal achievable* risk for the given  $p(\mathbf{x}, y)$  with any classifier!
- In a sense,  $R^*$  measures the inherent difficulty of the classification problem.

# Bayes risk

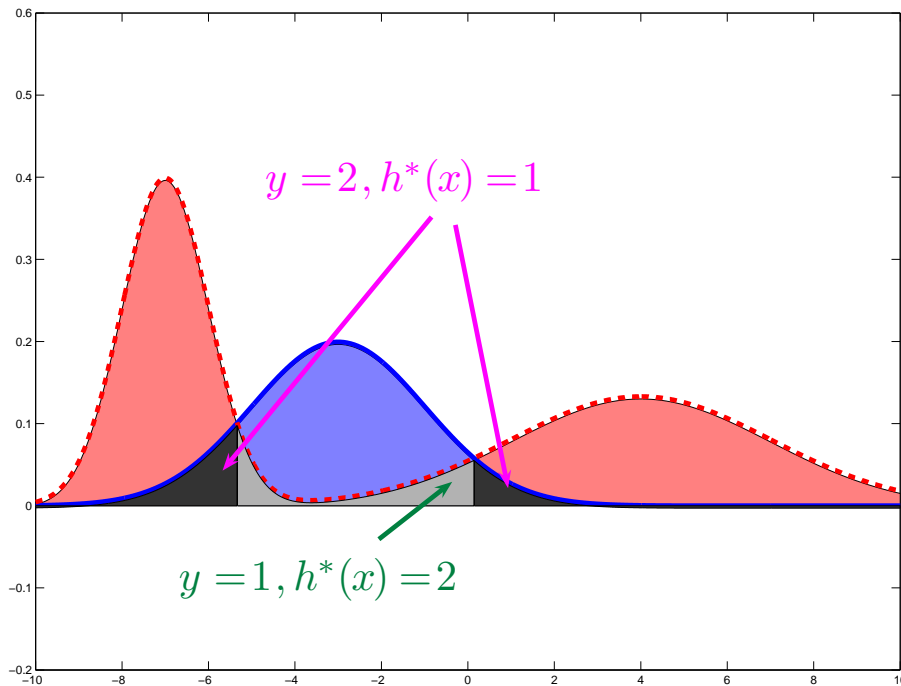


- The risk (probability of error) of Bayes classifier  $h^*$  is called the *Bayes risk*  $R^*$ .
- This is the *minimal achievable* risk for the given  $p(\mathbf{x}, y)$  with any classifier!
- In a sense,  $R^*$  measures the inherent difficulty of the classification problem.

- Easier to express in terms of probability of being correct:

$$R^* = 1 - \int_{\mathbf{x}} p(\mathbf{x}, y^*) d\mathbf{x}$$

# Bayes risk



- The risk (probability of error) of Bayes classifier  $h^*$  is called the *Bayes risk*  $R^*$ .
- This is the *minimal achievable* risk for the given  $p(\mathbf{x}, y)$  with any classifier!
- In a sense,  $R^*$  measures the inherent difficulty of the classification problem.

- Easier to express in terms of probability of being correct:

$$R^* = 1 - \int_{\mathbf{x}} \max_c \{p(\mathbf{x} | c = y) P_c\} d\mathbf{x}$$

---

# Discriminant function

- We can construct, for each class  $c$ , a *discriminant function*

$$\delta_c(\mathbf{x}) \triangleq \log p_c(\mathbf{x}) + \log P_c$$

such that

$$h^*(\mathbf{x}) = \operatorname{argmax}_c \delta_c(\mathbf{x}).$$

- We will always simplify  $\delta_c$  by removing terms and factors that are common for all  $\delta_c$  since they won't affect the decision boundary.
  - For example, if  $P_c = 1/C$  for all  $c$ , we can drop the prior term:

$$\delta_c(\mathbf{x}) = \log p_c(\mathbf{x})$$

---

## Two-category case

- In case of two classes  $y \in \{\pm 1\}$ , the Bayes classifier is

$$h^*(\mathbf{x}) = \operatorname{argmax}_{c=\pm 1} \delta_c(\mathbf{x}) = \operatorname{sign}(\delta_{+1}(\mathbf{x}) - \delta_{-1}(\mathbf{x})).$$

- Decision boundary is given by  $\delta_{+1}(\mathbf{x}) - \delta_{-1}(\mathbf{x}) = 0$ .
  - Sometimes  $f(\mathbf{x}) = \delta_{+1}(\mathbf{x}) - \delta_{-1}(\mathbf{x})$  is referred to as a discriminant function.
- With equal priors, this is equivalent to the *(log)-likelihood ratio test*:

$$h^*(\mathbf{x}) = \operatorname{sign} \left[ \log \frac{p(\mathbf{x} | y = +1)}{p(\mathbf{x} | y = -1)} \right].$$

---

# Linear discriminant functions

- When  $\delta_c$  are linear, the decision boundary is also linear.
- Example: class-conditionals are multivariate Gaussians with common covariance matrix

$$p_c(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \mu_c, \Sigma)$$

- As shown in Problem Apple-9,

$$\delta_c = \mu_c^T \Sigma^{-1} \mathbf{x} - \frac{1}{2} \mu_c^T \Sigma^{-1} \mu_c$$

---

# Linear discriminant functions

- When  $\delta_c$  are linear, the decision boundary is also linear.
- Example: class-conditionals are multivariate Gaussians with common covariance matrix

$$p_c(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \mu_c, \Sigma)$$

- As shown in Problem Apple-9,

$$\delta_c = \mu_c^T \Sigma^{-1} \mathbf{x} - \frac{1}{2} \mu_c^T \Sigma^{-1} \mu_c + \log P_c.$$

- This is a *linear* (in  $\mathbf{x}$ ) discriminant, thus the decision boundary is linear.

---

# Fisher's linear discriminant analysis revisited

- Assume two Gaussian class-conditionals, with equal covariances.
- The optimal decision boundary is

$$\begin{aligned}\delta_{+1}(\mathbf{x}) - \delta_{-1}(\mathbf{x}) &= (\mu_{+1} - \mu_{-1})^T \Sigma^{-1} \mathbf{x} - \frac{1}{2} \mu_{+1}^T \Sigma^{-1} \mu_{+1} + \frac{1}{2} \mu_{-1}^T \Sigma^{-1} \mu_{-1} \\ &+ \log P_{+1} - \log P_{-1} = 0,\end{aligned}$$

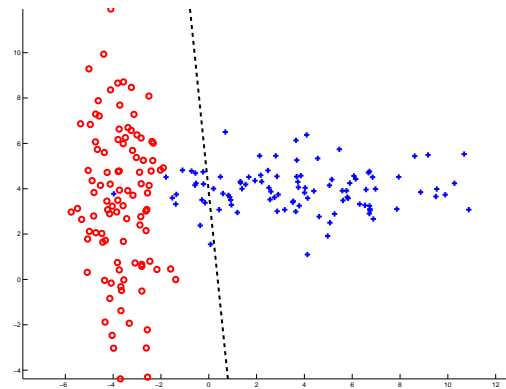
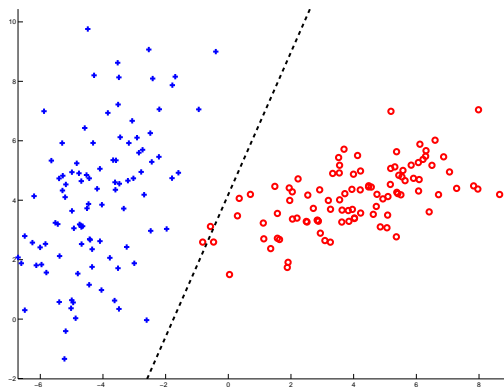
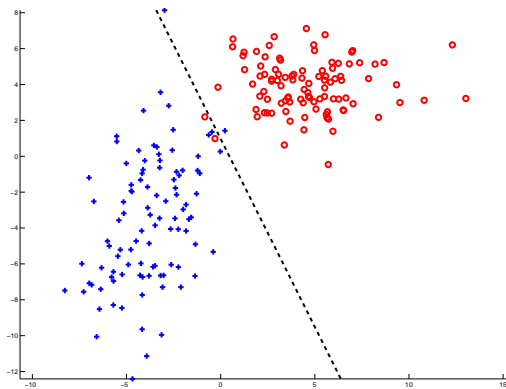
which is exactly the form we got for Fisher's LDA (plus we have a recipe for how to set  $w_0$ ).

– of course, instead of  $\mu_c, \Sigma$  in practice we use ML estimates  $\mathbf{m}_c, \sum_c N_c \mathbf{S}_c$ .

- So, under the assumption above, Fisher's LDA (with this choice of  $w_0$ ) is decision-theoretically optimal, up to estimation errors for means and covariance.

# Generative models for classification

- In generative models one explicitly models  $p(\mathbf{x}, y)$  or, equivalently,  $p_c(\mathbf{x})$  and  $P_c$ , to derive discriminants.
- Typically, the model imposes certain parametric form on the assumed distributions, and requires estimation of the parameters from data.
  - Most popular: Gaussian for continuous, multinomial for discrete.
  - We will see later in this class *non-parametric* models.
- Often, the classifier is OK even if data clearly don't conform to assumptions.



---

# Maximum likelihood density estimation

- Let  $X = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$  be a set of data points
  - no labels; in the current context  $X$  all come from class  $c$
- We assume parametric distribution model  $p(\mathbf{x}; \theta)$ .
- The (log)-likelihood of  $\theta$  given  $X$  (assuming i.i.d. sampling):

$$\ell(X; \theta) \triangleq \sum_{i=1}^N \log p(\mathbf{x}_i; \theta).$$

- ML estimate of  $\theta$ :

$$\hat{\theta}_{ML} \triangleq \operatorname{argmax}_{\theta} \ell(X; \theta)$$

- Intuitively: the observed data is most likely (has highest probability) for these settings of  $\theta$ .

---

## Gaussians with unequal covariances

- What if we remove the restriction that  $\forall c, \Sigma_c = \Sigma$ ?
- Compute ML estimate for  $\mu_c, \Sigma_c$  for each  $c$ .
- We get discriminants (and decision boundaries) *quadratic* in  $\mathbf{x}$ :

$$\delta_c(\mathbf{x}) = -\frac{1}{2}\mathbf{x}^T \Sigma_c^{-1} \mathbf{x} + \mu_c^T \Sigma_c^{-1} \mathbf{x} - \langle \text{const in } \mathbf{x} \rangle$$

(as shown in Problem Apple-10).

A quadratic form in  $\mathbf{x}$ :  $\mathbf{x}^T \mathbf{A} \mathbf{x}$ .

---

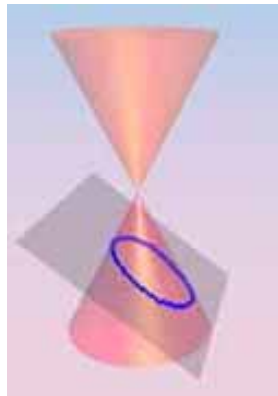
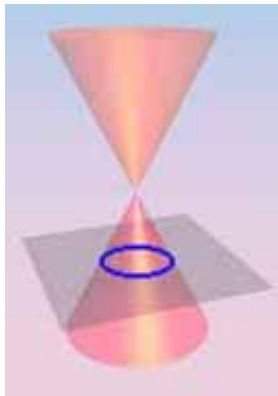
# Quadratic decision boundaries

- What do quadratic boundaries look like in 2D?

---

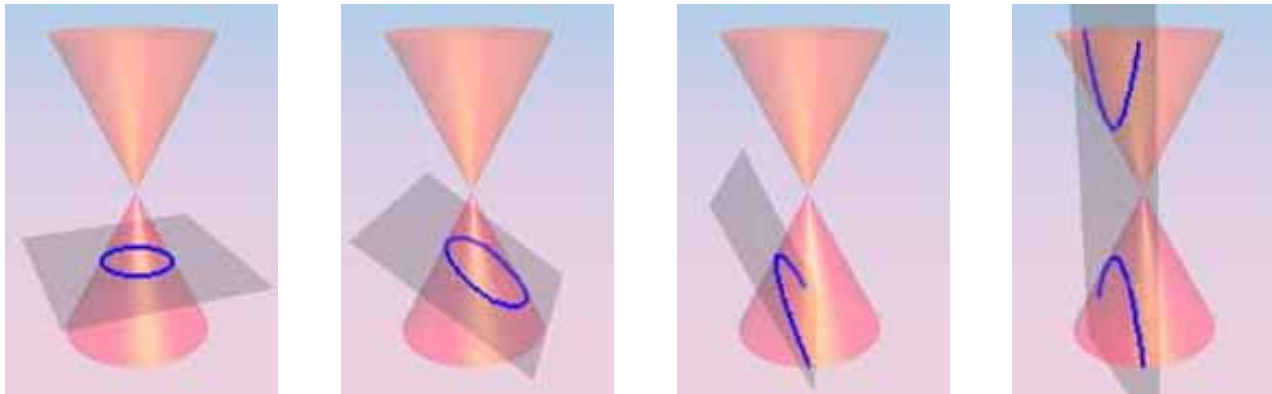
# Quadratic decision boundaries

- What do quadratic boundaries look like in 2D?
- Second-degree curves can be any *conic section*:



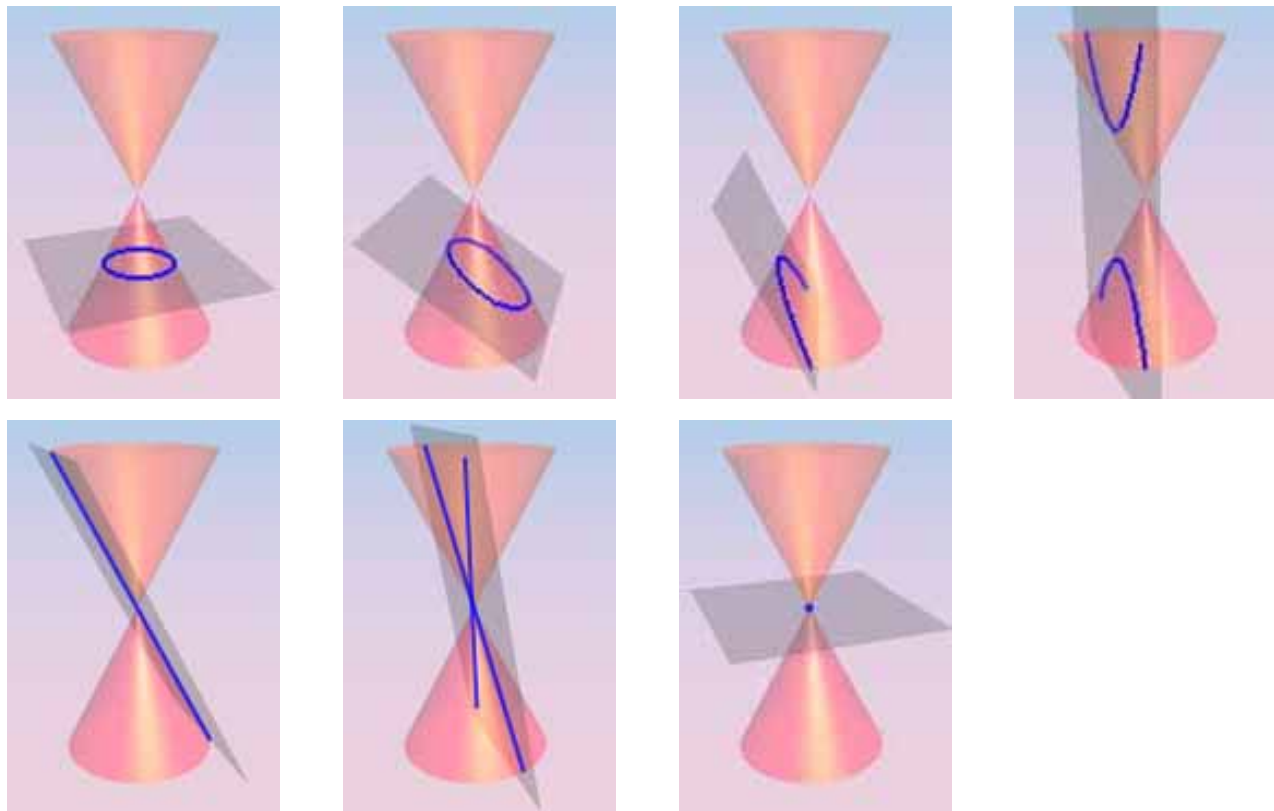
# Quadratic decision boundaries

- What do quadratic boundaries look like in 2D?
- Second-degree curves can be any *conic section*:



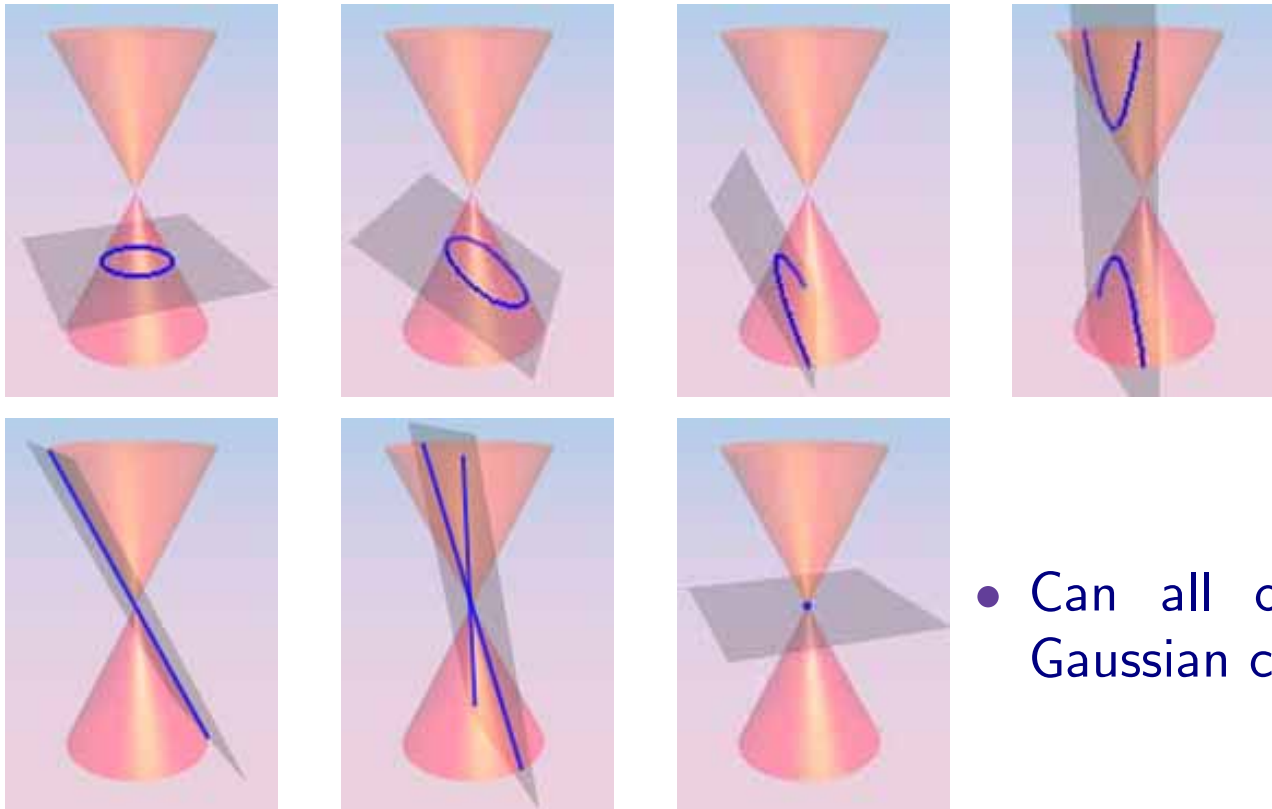
# Quadratic decision boundaries

- What do quadratic boundaries look like in 2D?
- Second-degree curves can be any *conic section*:



# Quadratic decision boundaries

- What do quadratic boundaries look like in 2D?
- Second-degree curves can be any *conic section*:



- Can all of these arise from two Gaussian classes?

---

## Next time

More on generative models.

Naïve Bayes classifiers.

Discrete data.