

# CSC 311: Introduction to Machine Learning

## Lecture 8 - Multivariate Gaussians, GDA

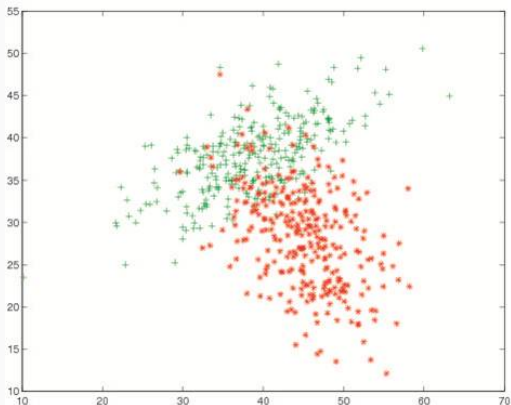
---

Amanjit Singh Kainth

University of Toronto, Summer 2026

# Classification: Diabetes Example

- Observation per patient: White blood cell count & glucose value.



- $p(\mathbf{x} | t = k)$  for each class is shaped like an ellipse  
⇒ we model each class as a multivariate Gaussian

# Overview

- Last week, we started our tour of probabilistic models, and introduced the fundamental concepts in the discrete setting.
- Continuous random variables:
  - ▶ Manipulating Gaussians to tackle interesting problems requires lots of linear algebra, so we'll begin with a linear algebra review.
    - ▶ Additional reference: See also Chapter 4 of Mathematics for Machine Learning, by Desienroth et al. <https://mml-book.github.io/>
- **Regression:** Linear regression as maximum likelihood estimation under a Gaussian distribution.
- **Generative classifier for continuous data:** Gaussian discriminant analysis, a Bayes classifier for continuous variables.
- Next week's lecture (PCA) draws heavily on today's linear algebra content, so be sure to review it offline.

# Linear Algebra Review

---

- 1 Linear Algebra Review
- 2 Multivariate Gaussian Distribution
- 3 Gaussian Maximum Likelihood
- 4 Revisiting Linear Regression
- 5 Gaussian Discriminant Analysis

# Eigenvectors and Eigenvalues

- Let  $\mathbf{B}$  be a square matrix.
- An **eigenvector** of  $\mathbf{B}$  is a *non-zero* vector  $\mathbf{v}$  such that

$$\mathbf{B}\mathbf{v} = \lambda\mathbf{v}$$

for a scalar  $\lambda$ , which is called an **eigenvalue**.

- A matrix of size  $D \times D$  has at most  $D$  distinct eigenvalues, but may have fewer.
- We will focus on symmetric matrices.

# Spectral Theorem

For a symmetric  $D \times D$  matrix,

- All of the eigenvalues are real-valued.
- There is a full set of  $D$  linearly independent eigenvectors. These eigenvectors form a basis for  $\mathbb{R}^D$ .
- The eigenvectors can be chosen to be real-valued.
- The eigenvectors can be chosen to be orthonormal.

# Spectral Decomposition

Factorize a symmetric matrix  $\mathbf{A}$  with the **Spectral Decomposition**:

$$\mathbf{A} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T$$

where

- $\mathbf{Q}$  is an orthogonal matrix
  - ▶ The columns  $\mathbf{q}_i$  of  $\mathbf{Q}$  are eigenvectors.
- $\mathbf{\Lambda}$  is a diagonal matrix.
  - ▶ The diagonal entries  $\lambda_i$  are the corresponding eigenvalues.

Check that this is reasonable:

$$\mathbf{A}\mathbf{q}_i =$$

# Spectral Decomposition

- Because  $\mathbf{A}$  has a full set of orthonormal eigenvectors  $\{\mathbf{q}_i\}$ , we can use these as an orthonormal basis for  $\mathbb{R}^D$ .
- A vector  $\mathbf{x}$  can be written in an alternate coordinate system:

$$\mathbf{x} = \tilde{x}_1 \mathbf{q}_1 + \cdots + \tilde{x}_D \mathbf{q}_D$$

- Converting between the two coordinate systems:

$$\tilde{\mathbf{x}} = \mathbf{Q}^\top \mathbf{x} \quad \mathbf{x} = \mathbf{Q} \tilde{\mathbf{x}}$$

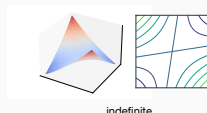
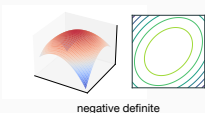
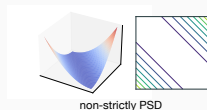
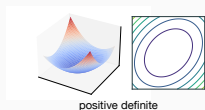
- In the alternate coordinate system,  $\mathbf{A}$  acts by re-scaling the individual coordinates:

$$\begin{aligned} \mathbf{A} \mathbf{x} &= \tilde{x}_1 \mathbf{A} \mathbf{q}_1 + \cdots + \tilde{x}_D \mathbf{A} \mathbf{q}_D \\ &= \lambda_1 \tilde{x}_1 \mathbf{q}_1 + \cdots + \lambda_D \tilde{x}_D \mathbf{q}_D \end{aligned}$$

# PSD Matrices

Symmetric matrices represent **quadratic forms**,  $f(\mathbf{v}) = \mathbf{v}^\top \mathbf{A} \mathbf{v}$ .

- If  $\mathbf{v}^\top \mathbf{A} \mathbf{v} > 0$  for all  $\mathbf{v} \neq \mathbf{0}$ ,  $\mathbf{A}$  is **positive definite**, denoted  $\mathbf{A} \succ \mathbf{0}$ .
- If  $\mathbf{v}^\top \mathbf{A} \mathbf{v} \geq 0$  for all  $\mathbf{v}$ ,  $\mathbf{A}$  is **positive semi-definite**, denoted  $\mathbf{A} \succeq \mathbf{0}$ .
- If  $\mathbf{v}^\top \mathbf{A} \mathbf{v} < 0$  for all  $\mathbf{v} \neq \mathbf{0}$ ,  $\mathbf{A}$  is **negative definite**, denoted  $\mathbf{A} \prec \mathbf{0}$ .
- If  $\mathbf{v}^\top \mathbf{A} \mathbf{v}$  can be positive or negative,  $\mathbf{A}$  is **indefinite**.



# PSD Matrices

- **Exercise:** Non-negative linear combinations of PSD matrices are PSD.
  
- **Related:** If  $\mathbf{A}$  is a random matrix which is always PSD, then  $\mathbb{E}[\mathbf{A}]$  is PSD.
- **Exercise:** For any matrix  $\mathbf{B}$ , the matrix  $\mathbf{B}\mathbf{B}^\top$  is PSD.
  
- **Corollary:** For a random vector  $\mathbf{x}$ , the **covariance matrix**  $\text{Cov}(\mathbf{x}) = \mathbf{E}[(\mathbf{x} - \boldsymbol{\mu})(\mathbf{x} - \boldsymbol{\mu})^\top]$  is a PSD matrix. (Special case of above, since  $\mathbf{x} - \boldsymbol{\mu}$  is a column vector, i.e. a  $D \times 1$  matrix.)

# PSD Matrices

**Claim:**  $\mathbf{A}$  is positive definite (PSD) if and only if all of its eigenvalues are positive (non-negative).

**Proof:** Write  $\mathbf{v}$  in terms of the eigenbases,

$$\tilde{\mathbf{v}} = \mathbf{Q}^T \mathbf{v}.$$

Then, we have

$$\begin{aligned} \mathbf{v}^T \mathbf{A} \mathbf{v} &= \mathbf{v}^T \mathbf{Q} \mathbf{\Lambda} \mathbf{Q}^T \mathbf{v} \\ &= \tilde{\mathbf{v}}^T \mathbf{\Lambda} \tilde{\mathbf{v}} \\ &= \sum_i \lambda_i \tilde{v}_i^2 \end{aligned}$$

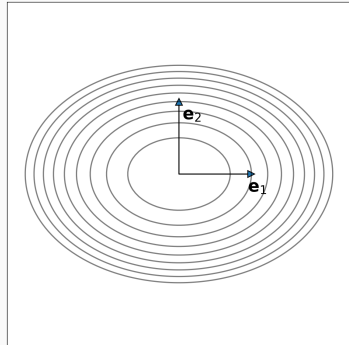
This is positive (nonnegative) for all  $\mathbf{v}$  if and only if all the  $\lambda_i$  are positive (nonnegative).

# PSD Matrices

- If  $\mathbf{A}$  is positive definite, then the contours of the quadratic form are elliptical.
- If  $\mathbf{A}$  is both diagonal and positive definite (i.e. its diagonal entries are positive), then the ellipses are axis-aligned.

$$\mathbf{A} = \begin{pmatrix} 0.5 & 0 \\ 0 & 1 \end{pmatrix}$$

$$\begin{aligned} f(\mathbf{v}) &= \mathbf{v}^\top \mathbf{A} \mathbf{v} \\ &= \sum_i a_i v_i^2 \end{aligned}$$

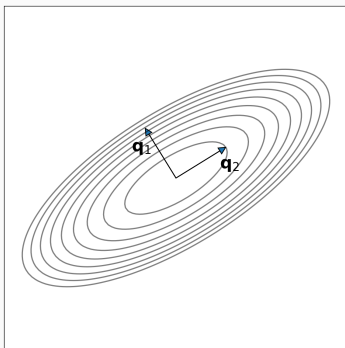


## PSD Matrices

For a positive definite  $\mathbf{A} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^\top$ , the contours of the quadratic form are elliptical, and the principal axes of the ellipses are aligned with the eigenvectors.

$$\mathbf{A} = \begin{pmatrix} 1 & -1 \\ -1 & 2 \end{pmatrix}$$

$$\begin{aligned} f(\mathbf{v}) &= \mathbf{v}^\top \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^\top \mathbf{v} \\ &= \tilde{\mathbf{v}}^\top \mathbf{\Lambda} \tilde{\mathbf{v}} \\ &= \sum_i \lambda_i \tilde{v}_i^2 \end{aligned}$$



In this example,  $\lambda_1 > \lambda_2$ .

All symmetric matrices are diagonal if you choose the right coordinate system.

# Matrix Powers

By the Spectral Decomposition, we can square a symmetric  $\mathbf{A}$ :

$$\mathbf{A}^2 = (\mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^\top)^2 = \mathbf{Q}\mathbf{\Lambda}\underbrace{\mathbf{Q}^\top\mathbf{Q}}_{=\mathbf{I}}\mathbf{\Lambda}\mathbf{Q}^\top = \mathbf{Q}\mathbf{\Lambda}^2\mathbf{Q}^\top$$

We can take the  $k$ -th power of  $\mathbf{A}$ :

$$\mathbf{A}^k = \mathbf{Q}\mathbf{\Lambda}^k\mathbf{Q}^\top.$$

If  $\mathbf{A}$  is invertible, we calculate its inverse:

$$\mathbf{A}^{-1} = (\mathbf{Q}^\top)^{-1}\mathbf{\Lambda}^{-1}\mathbf{Q}^{-1} = \mathbf{Q}\mathbf{\Lambda}^{-1}\mathbf{Q}^\top.$$

If  $\mathbf{A}$  is PSD, then we can calculate its **square root**:

$$\mathbf{A}^{1/2} = \mathbf{Q}\mathbf{\Lambda}^{1/2}\mathbf{Q}^\top.$$

# Determinant Properties

**Claim:** The determinant of a symmetric matrix equals the product of its eigenvalues.

$$|\mathbf{A}| = |\mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^\top| = |\mathbf{Q}||\mathbf{\Lambda}||\mathbf{Q}^\top| = |\mathbf{\Lambda}| = \prod_i \lambda_i.$$

**Corollary:** The determinant of a PSD (positive definite) matrix is non-negative (positive).

Basic properties of a determinant:

- $|\mathbf{BC}| = |\mathbf{B}| \cdot |\mathbf{C}|$
- $|\mathbf{B}| = 0$  iff  $\mathbf{B}$  is singular
- $|\mathbf{B}^{-1}| = |\mathbf{B}|^{-1}$  if  $\mathbf{B}$  is invertible (nonsingular)
- $|\mathbf{B}^\top| = |\mathbf{B}|$
- If  $\mathbf{Q}$  is orthogonal, then  $|\mathbf{Q}| = \pm 1$   
(i.e. orthogonal transformations preserve volume)
- If  $\mathbf{\Lambda}$  is diagonal with entries  $\{\lambda_i\}$ , then  $|\mathbf{\Lambda}| = \prod_i \lambda_i$ .

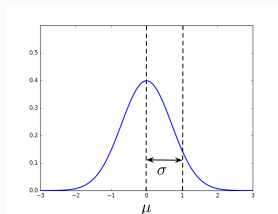
# Multivariate Gaussian Distribution

---

- 1 Linear Algebra Review
- 2 Multivariate Gaussian Distribution**
- 3 Gaussian Maximum Likelihood
- 4 Revisiting Linear Regression
- 5 Gaussian Discriminant Analysis

# Univariate Gaussian distribution

$$\mathcal{N}(x; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x - \mu)^2}{2\sigma^2}\right)$$



- Parameterized by mean  $\mu$  and variance  $\sigma^2$ .
- Why is Gaussian so popular?
  - ▶ Sums of lots of independent random variables are approximately Gaussian (Central Limit Theorem).
  - ▶ Machine learning uses Gaussians a lot because they make the calculations easy.

# Multivariate Mean and Covariance

Mean

$$\boldsymbol{\mu} = \mathbb{E}[\mathbf{x}] = \begin{pmatrix} \mu_1 \\ \vdots \\ \mu_D \end{pmatrix}$$

Covariance

$$\boldsymbol{\Sigma} = \text{Cov}(\mathbf{x}) = \mathbb{E}[(\mathbf{x} - \boldsymbol{\mu})(\mathbf{x} - \boldsymbol{\mu})^\top] = \begin{pmatrix} \sigma_1^2 & \sigma_{12} & \cdots & \sigma_{1D} \\ \sigma_{12} & \sigma_2^2 & \cdots & \sigma_{2D} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{D1} & \sigma_{D2} & \cdots & \sigma_D^2 \end{pmatrix}$$

( $\boldsymbol{\mu}$  and  $\boldsymbol{\Sigma}$ ) uniquely define a **multivariate Gaussian** (or **Normal**) distribution, denoted  $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$  or  $\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ .

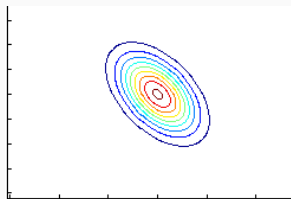
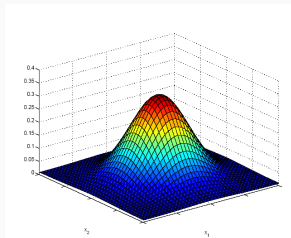
# PDF of Gaussian Distribution

PDF of the univariate Gaussian distribution ( $d = 1, \Sigma = \sigma^2$ ):

$$\mathcal{N}(x; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x - \mu)^2}{2\sigma^2}\right)$$

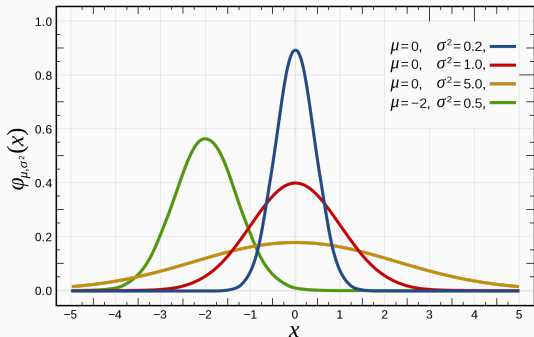
PDF of the multivariate Gaussian distribution:

$$\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{d/2} |\boldsymbol{\Sigma}|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu})\right]$$



# Univariate Shift + Scale

- All univariate Gaussian distributions are shaped like the **standard normal distribution**.
- Obtain  $\mathcal{N}(\mu, \sigma^2)$  by starting with  $\mathcal{N}(0, 1)$ , shifting by  $\mu$ , and stretching by  $\sigma = \sqrt{\sigma^2}$ .



## Multivariate Shift + Scale

- Start with a standard Gaussian  $\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ . So  $\mathbb{E}[\mathbf{x}] = \mathbf{0}$  and  $\text{Cov}(\mathbf{x}) = \mathbf{I}$ .
- What happens if we apply the map  $\hat{\mathbf{x}} = \mathbf{S}\mathbf{x} + \mathbf{b}$ ?
- By linearity of expectation,

$$\mathbb{E}[\hat{\mathbf{x}}] = \mathbf{S}\mathbb{E}[\mathbf{x}] + \mathbf{b} = \mathbf{b}.$$

- By the linear transformation rule for covariance,

$$\text{Cov}(\hat{\mathbf{x}}) = \mathbf{S} \text{Cov}(\mathbf{x}) \mathbf{S}^{\top} = \mathbf{S} \mathbf{S}^{\top}.$$

- $\hat{\mathbf{x}}$  is also Gaussian distributed.

$$\mathbb{E}[\mathbf{S}\mathbf{x} + \mathbf{b}] = \mathbf{b}$$

$$\text{Cov}(\mathbf{S}\mathbf{x} + \mathbf{b}) = \mathbf{S}\mathbf{S}^\top.$$

- To obtain  $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ , we start with  $\mathcal{N}(\mathbf{0}, \mathbf{I})$ , shift by  $\boldsymbol{\mu}$ , and scale by the matrix square root  $\boldsymbol{\Sigma}^{1/2}$ .
  - ▶ **Recall:**  $\boldsymbol{\Sigma}^{1/2} = \mathbf{Q}\boldsymbol{\Lambda}^{1/2}\mathbf{Q}$ .
  - ▶ For each eigenvector  $\mathbf{q}_i$  with eigenvalue  $\lambda_i$ , we stretch by a factor of  $\sqrt{\lambda_i}$  in the direction  $\mathbf{q}_i$ .

# Bivariate Gaussian

$$\Sigma = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$\Sigma = 0.5 \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$\Sigma = 2 \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

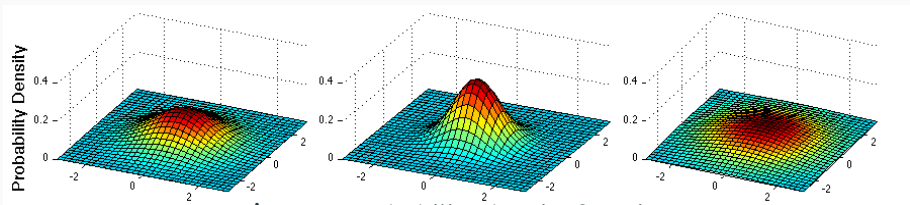


Figure 1: Probability density function

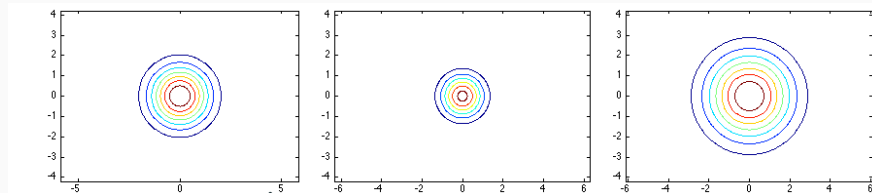


Figure 2: Contour plot of the pdf

# Bivariate Gaussian

$$\Sigma = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$\Sigma = \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix}$$

$$\Sigma = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}$$

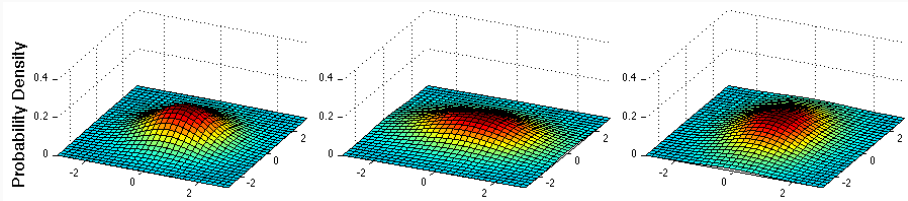


Figure 3: Probability density function

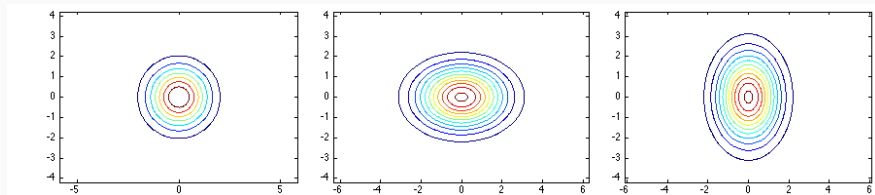


Figure 4: Contour plot of the pdf

# Bivariate Gaussian

$$\Sigma = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$\Sigma = \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix} \\ = \mathbf{Q}_1 \begin{pmatrix} 1.5 & 0. \\ 0. & 0.5 \end{pmatrix} \mathbf{Q}_1^T$$

$$\Sigma = \begin{pmatrix} 1 & 0.8 \\ 0.8 & 1 \end{pmatrix} \\ = \mathbf{Q}_2 \begin{pmatrix} 1.8 & 0. \\ 0. & 0.2 \end{pmatrix} \mathbf{Q}_2^T$$

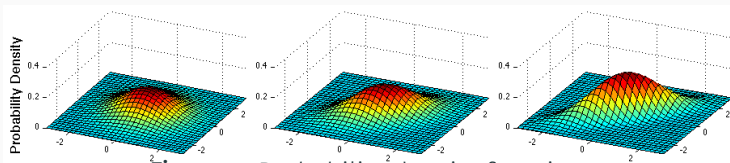


Figure 5: Probability density function

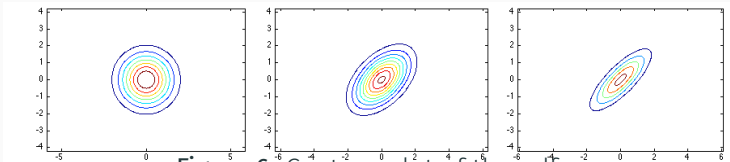


Figure 6: Contour plot of the pdf

# Gaussian Maximum Likelihood

---

- 1 Linear Algebra Review
- 2 Multivariate Gaussian Distribution
- 3 Gaussian Maximum Likelihood**
- 4 Revisiting Linear Regression
- 5 Gaussian Discriminant Analysis

## Maximum Likelihood for Multivariate Gaussian

Model the distribution of highest and lowest temperatures in Toronto in March, and recorded the following observations

$(-2.5, -7.5)$   $(-9.9, -14.9)$   $(-12.1, -17.5)$   $(-8.9, -13.9)$   $(-6.0, -11.1)$

Assume they're drawn from a Gaussian distribution  $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ .  
We want to estimate  $\boldsymbol{\mu}$  and  $\boldsymbol{\Sigma}$  using data.

## Maximum Likelihood for Univariate Gaussian

$$\frac{\partial \ell}{\partial \mu} = -\frac{1}{\sigma^2} \sum_{i=1}^N \mathbf{x}^{(i)} - \mu = 0$$

$$\hat{\mu}_{\text{ML}} = \frac{1}{N} \sum_{i=1}^N \mathbf{x}^{(i)}$$

## Maximum Likelihood for Univariate Gaussian

$$\begin{aligned}\frac{\partial \ell}{\partial \sigma} &= \frac{\partial}{\partial \sigma} \left[ \sum_{i=1}^N -\frac{1}{2} \log 2\pi - \log \sigma - \frac{1}{2\sigma^2} (\mathbf{x}^{(i)} - \mu)^2 \right] \\ &= \sum_{i=1}^N -\frac{1}{2} \frac{\partial}{\partial \sigma} \log 2\pi - \frac{\partial}{\partial \sigma} \log \sigma - \frac{\partial}{\partial \sigma} \frac{1}{2\sigma} (\mathbf{x}^{(i)} - \mu)^2 \\ &= \sum_{i=1}^N 0 - \frac{1}{\sigma} + \frac{1}{\sigma^3} (\mathbf{x}^{(i)} - \mu)^2 \\ &= -\frac{N}{\sigma} + \frac{1}{\sigma^3} \sum_{i=1}^N (\mathbf{x}^{(i)} - \mu)^2 = 0 \\ \hat{\sigma}_{\text{ML}} &= \sqrt{\frac{1}{N} \sum_{i=1}^N (\mathbf{x}^{(i)} - \mu)^2}\end{aligned}$$

# Maximum Likelihood for Multivariate Gaussian

Log-likelihood function:

$$\begin{aligned}\ell(\boldsymbol{\mu}, \boldsymbol{\Sigma}) &= \log \prod_{i=1}^N \left[ \frac{1}{(2\pi)^{d/2} |\boldsymbol{\Sigma}|^{1/2}} \exp \left\{ -\frac{1}{2} (\mathbf{x}^{(i)} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x}^{(i)} - \boldsymbol{\mu}) \right\} \right] \\ &= \sum_{i=1}^N \log \left[ \frac{1}{(2\pi)^{d/2} |\boldsymbol{\Sigma}|^{1/2}} \exp \left\{ -\frac{1}{2} (\mathbf{x}^{(i)} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x}^{(i)} - \boldsymbol{\mu}) \right\} \right] \\ &= \sum_{i=1}^N \underbrace{-\log(2\pi)^{d/2}}_{\text{constant}} - \log |\boldsymbol{\Sigma}|^{1/2} - \frac{1}{2} (\mathbf{x}^{(i)} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x}^{(i)} - \boldsymbol{\mu})\end{aligned}$$

# Gaussian Maximum Likelihood

Maximize the log-likelihood by setting the derivative to zero:

$$\begin{aligned}\frac{d\ell}{d\boldsymbol{\mu}} &= -\sum_{i=1}^N \frac{d}{d\boldsymbol{\mu}} \frac{1}{2} (\mathbf{x}^{(i)} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x}^{(i)} - \boldsymbol{\mu}) \\ &= -\sum_{i=1}^N \boldsymbol{\Sigma}^{-1} (\mathbf{x}^{(i)} - \boldsymbol{\mu}) = 0 \quad \text{using identity } \nabla_{\mathbf{x}} \mathbf{x}^T \mathbf{A} \mathbf{x} = 2\mathbf{A}\mathbf{x}\end{aligned}$$

Solving for  $\boldsymbol{\mu}$ , we get

$$\hat{\boldsymbol{\mu}} = \frac{1}{N} \sum_{i=1}^N \mathbf{x}^{(i)}.$$

The best estimate for  $\boldsymbol{\mu}$  is the sample mean of the observed values, or the **empirical mean**.

## Maximum Likelihood for Multivariate Gaussians

We can do a similar calculation for the covariance matrix  $\Sigma$ .

$$\begin{aligned}\frac{\partial \ell}{\partial \Sigma} &= 0 \\ \hat{\Sigma} &= \frac{1}{N} \sum_{i=1}^N (\mathbf{x}^{(i)} - \hat{\boldsymbol{\mu}})(\mathbf{x}^{(i)} - \hat{\boldsymbol{\mu}})^\top \\ &= \frac{1}{N} (\mathbf{X} - \mathbf{1}\boldsymbol{\mu}^\top)^\top (\mathbf{X} - \mathbf{1}\boldsymbol{\mu}^\top)\end{aligned}$$

where  $\mathbf{1}$  is an  $N$ -dimensional vector of 1s.

The best estimate for  $\Sigma$  is the **empirical covariance**.

# Revisiting Linear Regression

---

- 1 Linear Algebra Review
- 2 Multivariate Gaussian Distribution
- 3 Gaussian Maximum Likelihood
- 4 Revisiting Linear Regression**
- 5 Gaussian Discriminant Analysis

## Recap: Linear Regression

- Given a training set of inputs and targets  $\{(\mathbf{x}^{(i)}, t^{(i)})\}_{i=1}^N$
- Linear model:

$$y = \mathbf{w}^\top \mathbf{x}$$

- Squared error loss:

$$\mathcal{L}(y, t) = \frac{1}{2}(t - y)^2$$

- $L_2$  regularization:

$$\mathcal{R}(\mathbf{w}) = \frac{\lambda}{2} \|\mathbf{w}\|^2$$

- Closed-form solution:

$$\mathbf{w} = (\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^\top \mathbf{t}$$

- Gradient descent update rule:

$$\mathbf{w} \leftarrow (1 - \alpha\lambda)\mathbf{w} - \alpha \mathbf{X}^\top (\mathbf{y} - \mathbf{t})$$

# Linear Regression as Maximum Likelihood

- Let's give linear regression a probabilistic interpretation.
- Assume a Gaussian noise model.

$$t | \mathbf{x} \sim \mathcal{N}(\mathbf{w}^\top \mathbf{x}, \sigma^2)$$

- Linear regression is just maximum likelihood under this model:

$$\begin{aligned} \frac{1}{N} \sum_{i=1}^N \log p(t^{(i)} | \mathbf{x}^{(i)}; \mathbf{w}, b) &= \frac{1}{N} \sum_{i=1}^N \log \mathcal{N}(t^{(i)}; \mathbf{w}^\top \mathbf{x}, \sigma^2) \\ &= \frac{1}{N} \sum_{i=1}^N \log \left[ \frac{1}{\sqrt{2\pi}\sigma} \exp \left( -\frac{(t^{(i)} - \mathbf{w}^\top \mathbf{x})^2}{2\sigma^2} \right) \right] \\ &= \text{const} - \frac{1}{2N\sigma^2} \sum_{i=1}^N (t^{(i)} - \mathbf{w}^\top \mathbf{x})^2 \end{aligned}$$

# Regularization as MAP Inference

- View an  $L_2$  regularizer as MAP inference with a Gaussian prior.
- Recall MAP inference:

$$\arg \max_{\mathbf{w}} \log p(\mathbf{w} | \mathcal{D}) = \arg \max_{\mathbf{w}} [\log p(\mathbf{w}) + \log p(\mathcal{D} | \mathbf{w})]$$

- We just derived the likelihood term  $\log p(\mathcal{D} | \mathbf{w})$ :

$$\log p(\mathcal{D} | \mathbf{w}) = -\frac{1}{2N\sigma^2} \sum_{i=1}^N (t^{(i)} - \mathbf{w}^\top \mathbf{x}^{(i)})^2 + \text{const}$$

- Assume a Gaussian prior,  $\mathbf{w} \sim \mathcal{N}(\mathbf{m}, \mathbf{S})$ :

$$\begin{aligned} \log p(\mathbf{w}) &= \log \mathcal{N}(\mathbf{w}; \mathbf{m}, \mathbf{S}) \\ &= \log \left[ \frac{1}{(2\pi)^{D/2} |\mathbf{S}|^{1/2}} \exp \left( -\frac{1}{2} (\mathbf{w} - \mathbf{m})^\top \mathbf{S}^{-1} (\mathbf{w} - \mathbf{m}) \right) \right] \\ &= -\frac{1}{2} (\mathbf{w} - \mathbf{m})^\top \mathbf{S}^{-1} (\mathbf{w} - \mathbf{m}) + \text{const} \end{aligned}$$

- Commonly,  $\mathbf{m} = \mathbf{0}$  and  $\mathbf{S} = \eta \mathbf{I}$ , so

$$\log p(\mathbf{w}) = -\frac{1}{2\eta} \|\mathbf{w}\|^2 + \text{const}.$$

This is just  $L_2$  regularization!

# Gaussian Discriminant Analysis

---

- 1 Linear Algebra Review
- 2 Multivariate Gaussian Distribution
- 3 Gaussian Maximum Likelihood
- 4 Revisiting Linear Regression
- 5 Gaussian Discriminant Analysis**

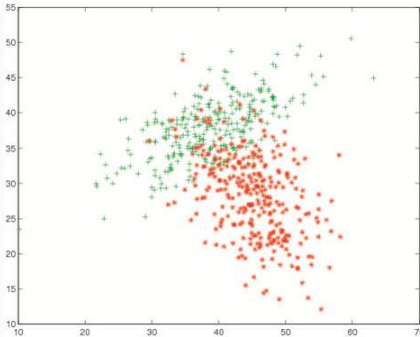
# Generative vs Discriminative (Recap)

Two approaches to classification:

- **Discriminative approach:** estimate parameters of decision boundary/class separator directly from labeled examples.
  - ▶ Model  $p(t|\mathbf{x})$  directly (logistic regression models)
  - ▶ Learn mappings from inputs to classes (linear/logistic regression, decision trees etc)
  - ▶ Tries to solve: How do I separate the classes?
- **Generative approach:** model the distribution of inputs characteristic of the class (Bayes classifier).
  - ▶ Model  $p(\mathbf{x}|t)$
  - ▶ Apply Bayes Rule to derive  $p(t|\mathbf{x})$ .
  - ▶ Tries to solve: What does each class "look" like?

# Classification: Diabetes Example

- Gaussian discriminant analysis (GDA) is a Bayes classifier for continuous-valued inputs.
- Observation per patient: White blood cell count & glucose value.



- $p(\mathbf{x} | t = k)$  for each class is shaped like an ellipse  
⇒ we model each class as a multivariate Gaussian 38

# Gaussian Discriminant Analysis

- **Gaussian Discriminant Analysis** in its general form assumes that  $p(\mathbf{x}|t)$  is distributed according to a multivariate Gaussian distribution
- Multivariate Gaussian distribution:

$$p(\mathbf{x} | t = k) = \frac{1}{(2\pi)^{D/2} |\boldsymbol{\Sigma}_k|^{1/2}} \exp \left[ -\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1} (\mathbf{x} - \boldsymbol{\mu}_k) \right]$$

where  $|\boldsymbol{\Sigma}_k|$  denotes the determinant of the matrix.

- Each class  $k$  has associated mean vector  $\boldsymbol{\mu}_k$  and covariance matrix  $\boldsymbol{\Sigma}_k$
- How many parameters?
  - ▶ Each  $\boldsymbol{\mu}_k$  has  $D$  parameters, for  $DK$  total.
  - ▶ Each  $\boldsymbol{\Sigma}_k$  has  $\mathcal{O}(D^2)$  parameters, for  $\mathcal{O}(D^2K)$  — could be hard to estimate (more on that later).

# GDA: Learning

- Learn the parameters for each class using maximum likelihood
- For simplicity, assume binary classification

$$p(t | \phi) = \phi^t (1 - \phi)^{1-t}$$

- You can compute the ML estimates in closed form ( $\phi$  and  $\boldsymbol{\mu}_k$  are easy,  $\boldsymbol{\Sigma}_k$  is tricky)

$$\phi = \frac{1}{N} \sum_{i=1}^N r_1^{(i)}$$

$$\boldsymbol{\mu}_k = \frac{\sum_{i=1}^N r_k^{(i)} \cdot \mathbf{x}^{(i)}}{\sum_{i=1}^N r_k^{(i)}}$$

$$\boldsymbol{\Sigma}_k = \frac{1}{\sum_{i=1}^N r_k^{(i)}} \sum_{i=1}^N r_k^{(i)} (\mathbf{x}^{(i)} - \boldsymbol{\mu}_k)(\mathbf{x}^{(i)} - \boldsymbol{\mu}_k)^\top$$

$$r_k^{(i)} = \mathbb{1}[t^{(i)} = k]$$

# GDA Decision Boundary

- Recall: for Bayes classifiers, we compute the decision boundary with Bayes' Rule:

$$p(t | \mathbf{x}) = \frac{p(t) p(\mathbf{x} | t)}{\sum_{t'} p(t') p(\mathbf{x} | t')}$$

- Plug in the Gaussian  $p(\mathbf{x} | t)$ :

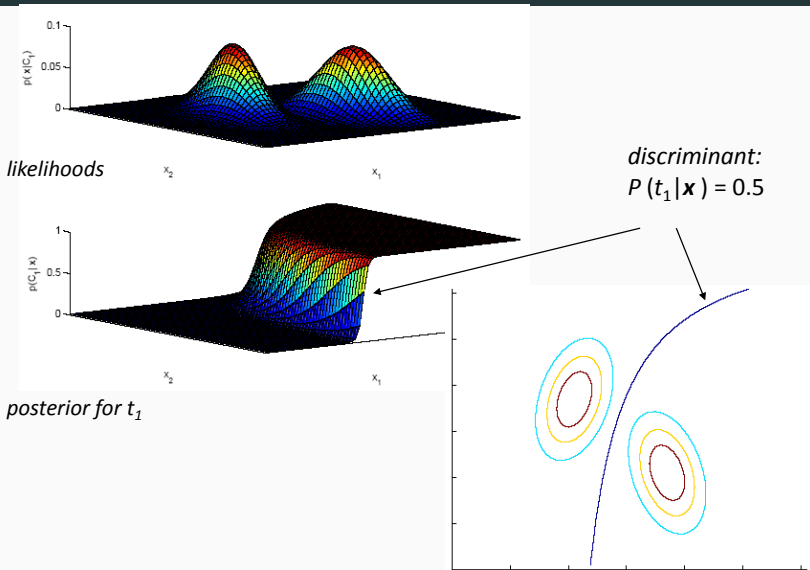
$$\begin{aligned} \log p(t_k | \mathbf{x}) &= \log p(\mathbf{x} | t_k) + \log p(t_k) - \log p(\mathbf{x}) \\ &= -\frac{D}{2} \log(2\pi) - \frac{1}{2} \log |\boldsymbol{\Sigma}_k| - \frac{1}{2} (\mathbf{x} - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1} (\mathbf{x} - \boldsymbol{\mu}_k) + \\ &\quad + \log p(t_k) - \log p(\mathbf{x}) \end{aligned}$$

- Decision boundary:

$$(\mathbf{x} - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1} (\mathbf{x} - \boldsymbol{\mu}_k) = (\mathbf{x} - \boldsymbol{\mu}_\ell)^\top \boldsymbol{\Sigma}_\ell^{-1} (\mathbf{x} - \boldsymbol{\mu}_\ell) + \text{Const}$$

- What's the shape of the boundary?
  - ▶ We have a quadratic function in  $\mathbf{x}$ , so the decision boundary is a conic section!

# GDA Decision Boundary



# GDA Decision Boundary

- Our equation for the decision boundary:

$$(\mathbf{x} - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1} (\mathbf{x} - \boldsymbol{\mu}_k) = (\mathbf{x} - \boldsymbol{\mu}_\ell)^\top \boldsymbol{\Sigma}_\ell^{-1} (\mathbf{x} - \boldsymbol{\mu}_\ell) + \text{Const}$$

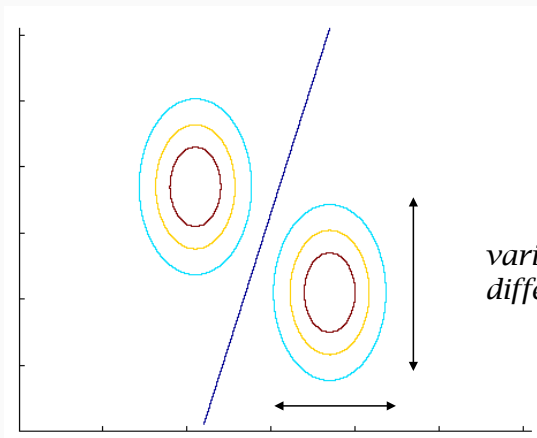
- Expand the product and factor out constants (w.r.t.  $\mathbf{x}$ ):

$$\mathbf{x}^\top \boldsymbol{\Sigma}_k^{-1} \mathbf{x} - 2\boldsymbol{\mu}_k^\top \boldsymbol{\Sigma}_k^{-1} \mathbf{x} = \mathbf{x}^\top \boldsymbol{\Sigma}_\ell^{-1} \mathbf{x} - 2\boldsymbol{\mu}_\ell^\top \boldsymbol{\Sigma}_\ell^{-1} \mathbf{x} + \text{Const}$$

- What if all classes share the same covariance  $\boldsymbol{\Sigma}$ ?
  - ▶ We get a linear decision boundary!

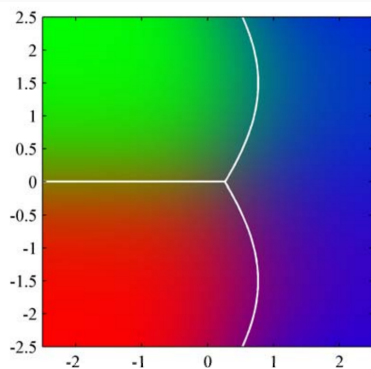
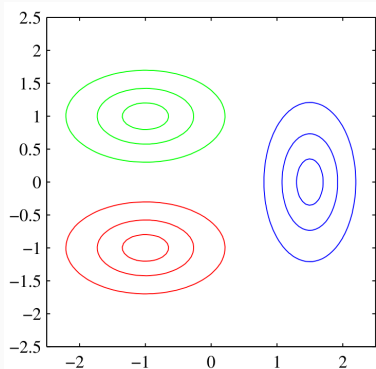
$$\begin{aligned} -2\boldsymbol{\mu}_k^\top \boldsymbol{\Sigma}^{-1} \mathbf{x} &= -2\boldsymbol{\mu}_\ell^\top \boldsymbol{\Sigma}^{-1} \mathbf{x} + \text{Const} \\ (\boldsymbol{\mu}_k - \boldsymbol{\mu}_\ell)^\top \boldsymbol{\Sigma}^{-1} \mathbf{x} &= \text{Const} \end{aligned}$$

## GDA Decision Boundary: Shared Covariances



*variances may be different*

## GDA Decision Boundary: Linear vs Quadratic



Classes that share the same covariance have linear decision boundaries. With classes that don't share the same covariance, the decision boundary is necessarily non-linear (quadratic discriminant).

Source: PRML Bishop, 2006

## GDA vs Logistic Regression

- Binary classification: If you examine  $p(t = 1 | \mathbf{x})$  under GDA and assume  $\Sigma_0 = \Sigma_1 = \Sigma$ , you will find that it looks like this:

$$p(t | \mathbf{x}, \phi, \boldsymbol{\mu}_0, \boldsymbol{\mu}_1, \boldsymbol{\Sigma}) = \frac{1}{1 + \exp(-\mathbf{w}^T \mathbf{x} - b)}$$

where  $(\mathbf{w}, b)$  are chosen based on  $(\phi, \boldsymbol{\mu}_0, \boldsymbol{\mu}_1, \boldsymbol{\Sigma})$ .

- Same model as logistic regression!

# GDA vs Logistic Regression

When should we prefer GDA to logistic regression, and vice versa?

- GDA makes a stronger modeling assumption: assumes class-conditional data is multivariate Gaussian
  - ▶ If this is true, GDA is asymptotically efficient (best model in limit of large  $N$ )
  - ▶ If it's not true, the quality of the predictions might suffer.
- Many class-conditional distributions lead to logistic classifier.
  - ▶ When these distributions are non-Gaussian (i.e., almost always), LR usually beats GDA
- GDA can handle easily missing features (how do you do that with LR?)

# Gaussian Naive Bayes

- What if  $\mathbf{x}$  is high-dimensional?
  - ▶ The  $\Sigma_k$  have  $\mathcal{O}(D^2K)$  parameters, which can be a problem if  $D$  is large.
  - ▶ We already saw we can save some a factor of  $K$  by using a shared covariance for the classes.
  - ▶ Any other idea you can think of?
- **Naive Bayes:** Assumes features independent given the class

$$p(\mathbf{x} | t = k) = \prod_{j=1}^D p(x_j | t = k)$$

- Assuming likelihoods are Gaussian, how many parameters required for Naive Bayes classifier?
  - ▶ This is equivalent to assuming the  $x_j$  are uncorrelated, i.e.  $\Sigma$  is diagonal.
  - ▶ Hence, only  $D$  parameters for  $\Sigma$ !

# Gaussian Naïve Bayes

- **Gaussian Naïve Bayes** classifier assumes that the likelihoods are Gaussian:

$$p(x_j | t = k) = \frac{1}{\sqrt{2\pi}\sigma_{jk}} \exp \left[ \frac{-(x_j - \mu_{jk})^2}{2\sigma_{jk}^2} \right]$$

(this is just a 1-dim Gaussian, one for each input dimension)

- Model the same as GDA with diagonal covariance matrix
- Maximum likelihood estimate of parameters

$$\mu_{jk} = \frac{\sum_{i=1}^N r_k^{(i)} x_j^{(i)}}{\sum_{i=1}^N r_k^{(i)}}$$

$$\sigma_{jk}^2 = \frac{\sum_{i=1}^N r_k^{(i)} (x_j^{(i)} - \mu_{jk})^2}{\sum_{i=1}^N r_k^{(i)}}$$

$$r_k^{(i)} = \mathbb{1}[t^{(i)} = k]$$

## Decision Boundary: Isotropic

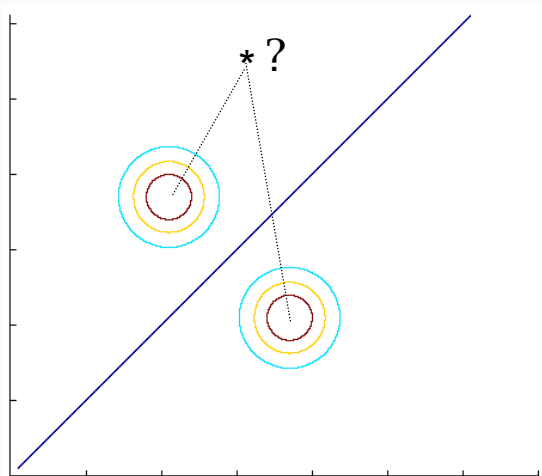
- We can go even further and assume the covariances are **spherical**, or **isotropic**.
- In this case:  $\Sigma = \sigma^2 \mathbf{I}$  (just need one parameter!)
- Going back to the class posterior for GDA:

$$\begin{aligned}\log p(t_k | \mathbf{x}) &= \log p(\mathbf{x} | t_k) + \log p(t_k) - \log p(\mathbf{x}) \\ &= -\frac{D}{2} \log(2\pi) - \frac{1}{2} \log |\Sigma_k^{-1}| - \frac{1}{2} (\mathbf{x} - \boldsymbol{\mu}_k)^\top \Sigma_k^{-1} (\mathbf{x} - \boldsymbol{\mu}_k) + \\ &\quad + \log p(t_k) - \log p(\mathbf{x})\end{aligned}$$

- Suppose for simplicity that  $p(t)$  is uniform. Plugging in  $\Sigma = \sigma^2 \mathbf{I}$  and simplifying a bit,

$$\begin{aligned}\log p(t_k | \mathbf{x}) - \log p(t_\ell | \mathbf{x}) &= -\frac{1}{2\sigma^2} [(\mathbf{x} - \boldsymbol{\mu}_k)^\top (\mathbf{x} - \boldsymbol{\mu}_k) - (\mathbf{x} - \boldsymbol{\mu}_\ell)^\top (\mathbf{x} - \boldsymbol{\mu}_\ell)] \\ &= -\frac{1}{2\sigma^2} [\|\mathbf{x} - \boldsymbol{\mu}_k\|^2 - \|\mathbf{x} - \boldsymbol{\mu}_\ell\|^2]\end{aligned}$$

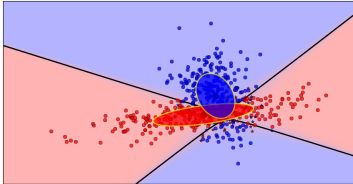
## Decision Boundary: Isotropic



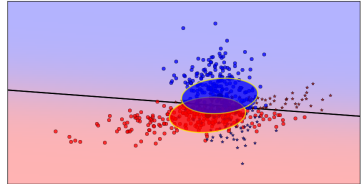
- The decision boundary bisects the class means!

# Example

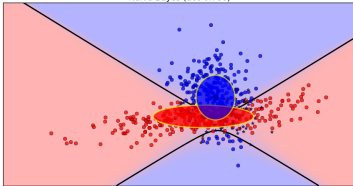
Full Covariances (acc 0.805)



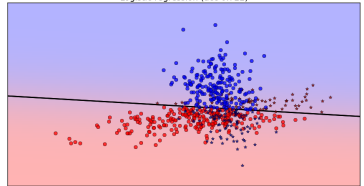
Shared Covariance (acc 0.717)



Naive Bayes (acc 0.780)



Logistic regression (acc 0.722)



## Generative models - Recap

- GDA has quadratic (conic) decision boundary.
- With shared covariance, GDA is similar to logistic regression.
- Generative models:
  - ▶ Flexible models, easy to add/remove class.
  - ▶ Handle missing data naturally.
  - ▶ More “natural” way to think about things, but usually doesn’t work as well.
- Tries to solve a hard problem (model  $p(\mathbf{x})$ ) in order to solve a easy problem (model  $p(t | \mathbf{x})$ ).

**Next up:** Unsupervised learning with PCA!