CSC 311: Introduction to Machine Learning

Lecture 8 - Multivariate Gaussians, GDA

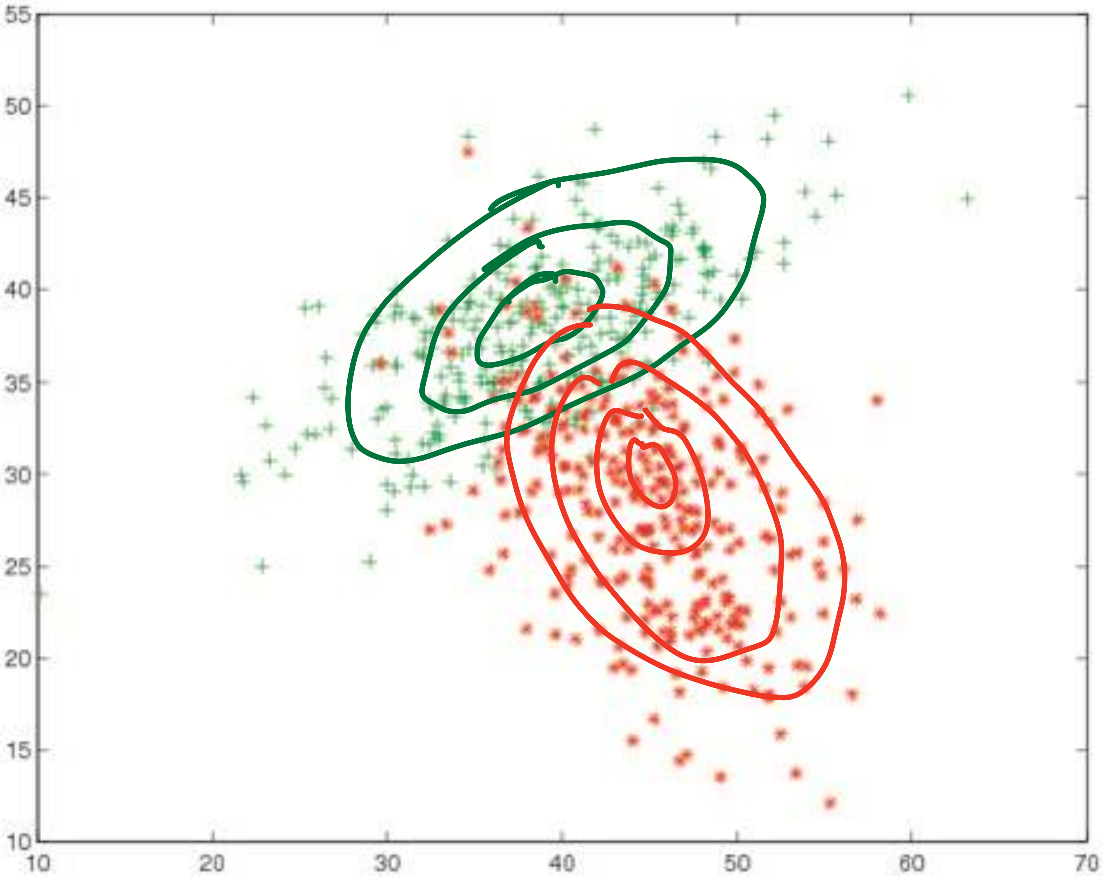
Michael Zhang Chandra Gummaluru

University of Toronto, Winter 2023

Classification: Diabetes Example

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

glucose



generative
model

wBC

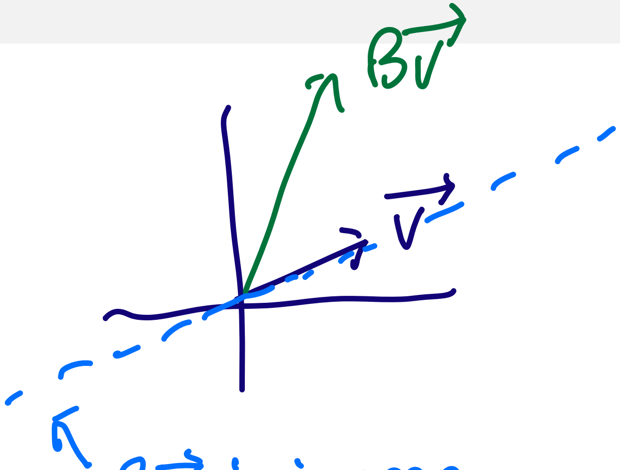
- $p(\mathbf{x} \,|\, t = k)$ for each class is shaped like an ellipse

$\Longrightarrow$ 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.

Eigenvectors and Eigenvalues



$B\vec{v}$ is in span if eigenvector

(linear transformation)

- Let **B** be a square matrix.
- An eigenvector of **B** is a vector **v** such that

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

$$(B-\lambda I)v=0$$

  for a scalar λ, 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

$V = [v_1 \; v_2 \; \dots \; v_D]$

$V^T V = I$

$\nearrow$ $v_1, v_2, \dots v_D$

each of these are eigenvectors

$v_i^T v_j = 0 \quad i \neq j$

$v_i^T v_i = 1$

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 **A** with the Spectral Decomposition:

$$Q^T = \begin{bmatrix} - & q_1 & - \\ - & q_2 & - \\ & \vdots & \\ - & q_n & - \end{bmatrix}$$

$$\mathbf{A} = \mathbf{Q}\boldsymbol{\Lambda}\mathbf{Q}^\top$$

$$\begin{bmatrix} q_1 & q_2 & \dots & q_n \end{bmatrix} \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_n \end{bmatrix} Q^T$$

where

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

$$Q^TQ = I$$

Check that this is reasonable:

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

$$AQ = Q\Lambda$$

$$A = AQQ^T = Q\Lambda Q^T$$

$$A \begin{bmatrix} q_1 & q_2 & \dots & q_n \end{bmatrix} = \begin{bmatrix} q_1 & q_2 & \dots & q_n \end{bmatrix} \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{bmatrix}$$

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:

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

Converts to eigenbasis   converts to original basis

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

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

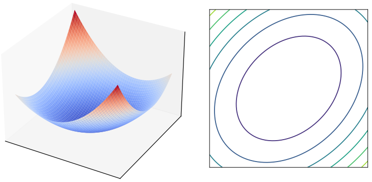
PSD Matrices

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

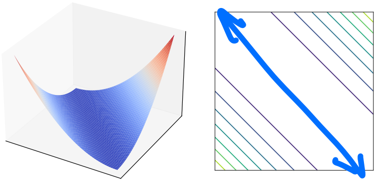
$>$

$\succ$

- 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.



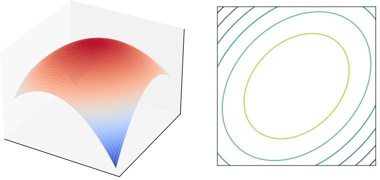
positive definite



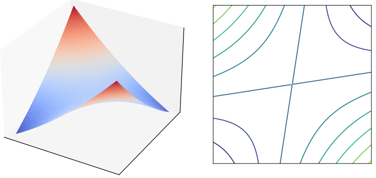
non-strictly PSD

PSD

$f(v)=0$



negative definite



indefinite

PSD Matrices

What are the eigenvectors of eigenvalues for A^{-1}?

- **Exercise:** Non-negative linear combinations of PSD matrices are PSD.

$A \succeq 0$ and $B \succeq 0$

prove $A + B \succeq 0$

$$\forall x \quad x^T A x \geq 0$$
$$x^T B x \geq 0$$

$$x^T (A+B) x$$
$$= x^T A x + x^T B x$$
$$\geq 0$$

- **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.

consider arbitrary x

$$(Ab)^T = b^T A^T$$

$$x^T B B^T x = (B^T x)^T (B^T x) = \|B^T x\|_2^2 \geq 0$$

$$c^T c = \|c\|_2^2$$

- **Corollary:** For a random vector $\mathbf{x}$, the covariance matrix $\mathrm{Cov}(\mathbf{x}) = \mathbf{E}[(\mathbf{x} - \boldsymbol{\mu})(\mathbf{x} - \boldsymbol{\mu})^\top]$ is a PSD matrix. (Special case of above, since $\mathbf{x} - \boldsymbol{\mu}$ is a column vector, i.e. a $D \times 1$ matrix.)

$$Av = \lambda v$$

$$A^{-1}Av = A^{-1}\lambda v$$

$$v = \lambda A^{-1} v$$

$$\frac{1}{\lambda} v = A^{-1} v$$

A^{-1} eigenvals $\quad \frac{1}{\lambda_1} \quad \frac{1}{\lambda_2} \quad \ldots \quad \frac{1}{\lambda_n}$

$\quad v_1 \quad v_2 \quad \ldots \quad v_n$

PSD Matrices

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

$$v^T A v$$

$$= \underbrace{v^T Q}_{w^T} \Lambda \underbrace{Q^T v}_{w}$$

$$= \sum_{i=1}^{n} \lambda_i w_i^2 \geq 0$$

$$\Longleftrightarrow$$

$$\lambda_i \geq 0$$

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

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

Then, we have

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

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

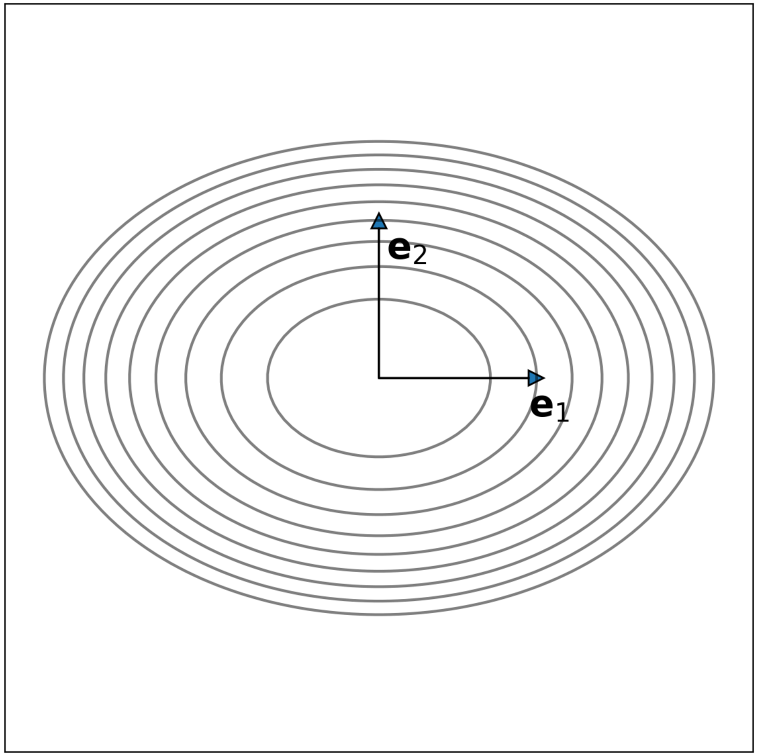
PSD Matrices

- 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.

$0.5x_1^2 + x_2^2$

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

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

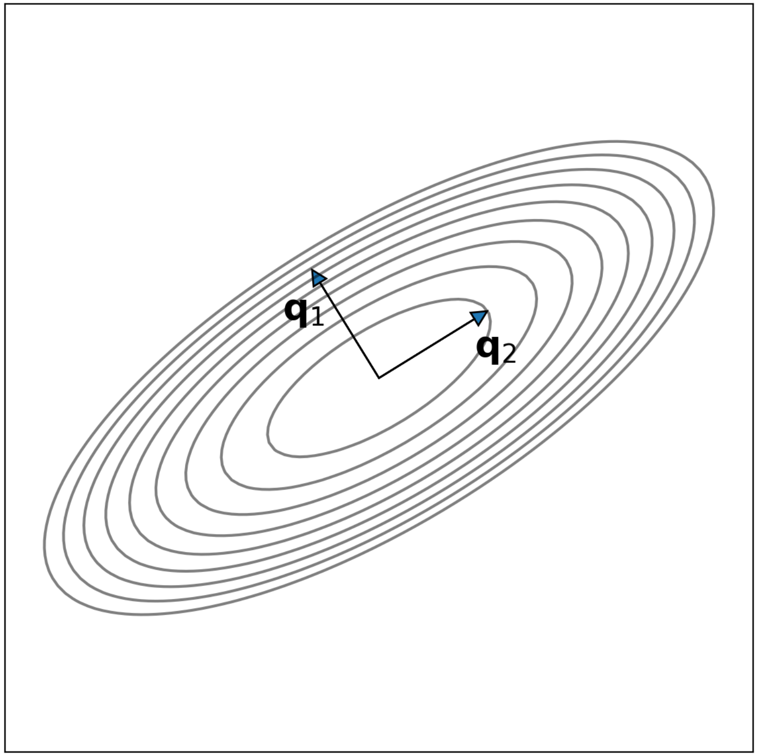


PSD Matrices

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 **A**:

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

↑

A^{100} really hard generally

$A^2 x$

Geometric interpretation

shift to eigenbasis
scale eigenvalues
shifts back] I
shift to
scale eigenvalues
shift back

We can take the k-th power of **A**:

$A^{100} = Q \Lambda^{100} Q^T$

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

If **A** is invertible, we calculate its inverse:

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

If **A** is PSD, then we can calculate its square root:

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

$(A^{\frac{1}{2}})(A^{\frac{1}{2}}) = A$

Determinant Properties

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

$$|\mathbf{A}| = |\mathbf{Q}\boldsymbol{\Lambda}\mathbf{Q}^\top| = \underbrace{|\mathbf{Q}||\boldsymbol{\Lambda}||\mathbf{Q}^\top|}_{|\Lambda||Q^TQ| = |\Lambda| 1} = |\boldsymbol{\Lambda}| = \prod_i \lambda_i.$$

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 $\boldsymbol{\Lambda}$ is diagonal with entries $\{\lambda_i\}$, then $|\boldsymbol{\Lambda}| = \prod_i \lambda_i$.

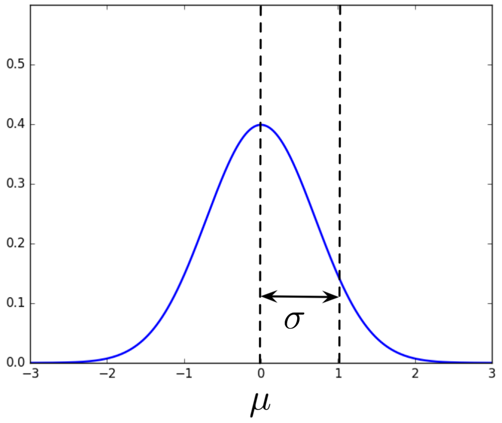
1. Linear Algebra Review
2. Multivariate Gaussian Distribution
3. Gaussian Maximum Likelihood
4. Revisiting Linear Regression
5. Gaussian Discriminant Analysis

Univariate Gaussian distribution

∝ probabability density
maximized at μ

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

$\sqrt{2\pi\sigma^2}$



- Parameterized by mean μ and variance σ^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

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

$E[(x_i - \mu_i)(x_j - \mu_j)]$
$= cov(x_i, x_j)$

Covariance

$E[(x_1 - \mu_1)(x_1 - \mu_1)] = var(x_1)$

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

($\boldsymbol{\mu}$ 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 ($d = 1$, $\boldsymbol{\Sigma} = \sigma^2$):

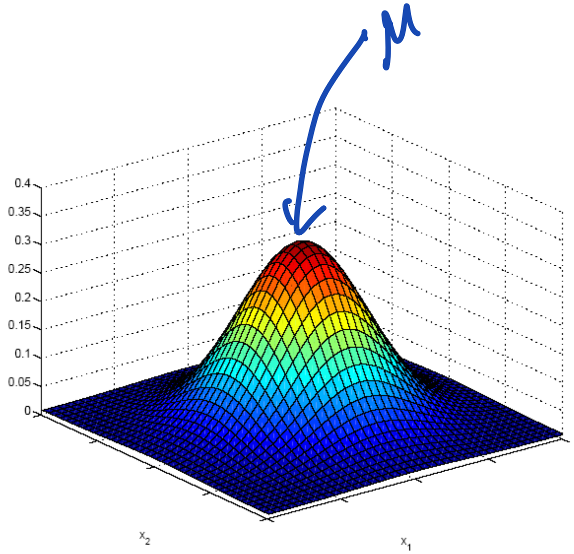
$$\mathcal{N}(x;\mu,\sigma^2) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

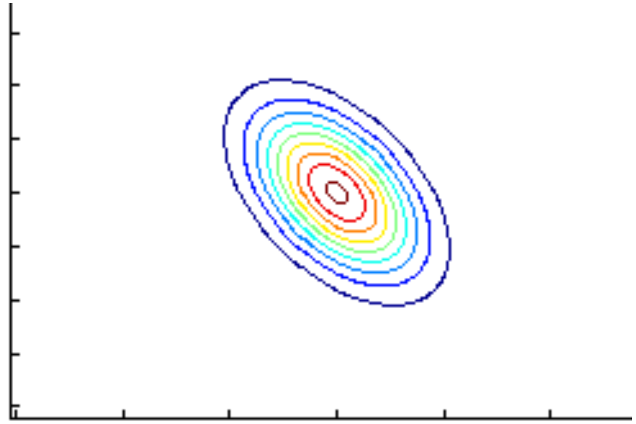
PDF of the multivariate Gaussian distribution:

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

$n=1$

$$-\frac{(x-\mu)^2}{2\sigma^2}$$





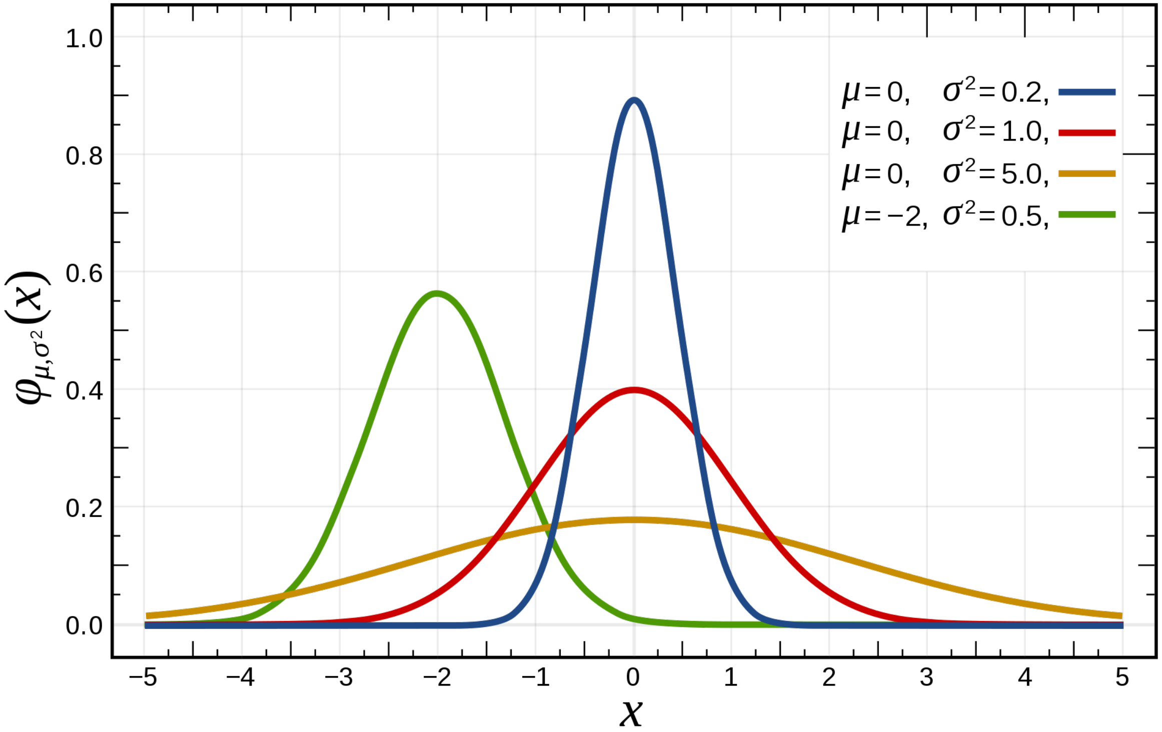
Univariate Shift + Scale

$z = \frac{x - \mu}{\sigma}$

- 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 μ, and stretching by $\sigma = \sqrt{\sigma^2}$.

σ^2 variance

σ standard dev.



scale the variance

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 $\mathrm{Cov}(\mathbf{x}) = \mathbf{I}$.
- What happens if we apply the map $\hat{\mathbf{x}} = \mathbf{S}\mathbf{x} + \mathbf{b}$?
- By linearity of expecation,

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

- By the linear transformation rule for covariance,

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

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

Multivariate Shift + Scale

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

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

- 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 λ_i, we stretch by a factor of $\sqrt{\lambda_i}$ in the direction $\mathbf{q}_i$.

Bivariate Gaussian

$$\boldsymbol{\Sigma} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \qquad \boldsymbol{\Sigma} = 0.5 \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \qquad \boldsymbol{\Sigma} = 2 \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

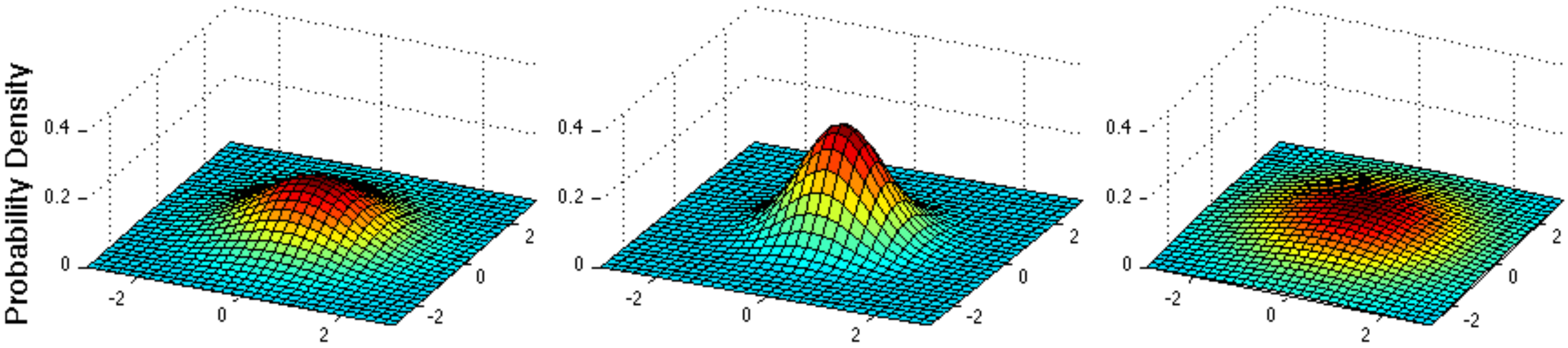


Figure: Probability density function

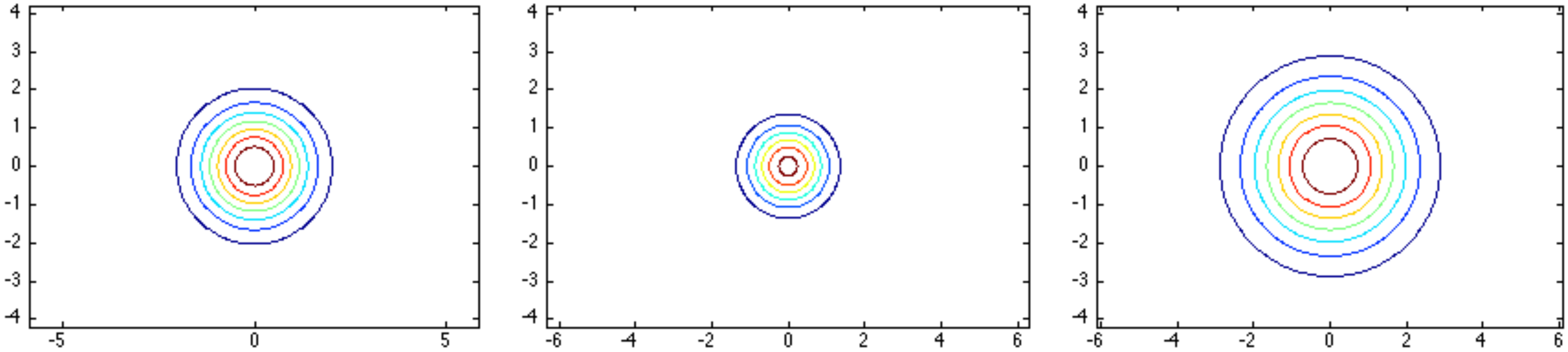


Figure: Contour plot of the pdf

Bivariate Gaussian

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

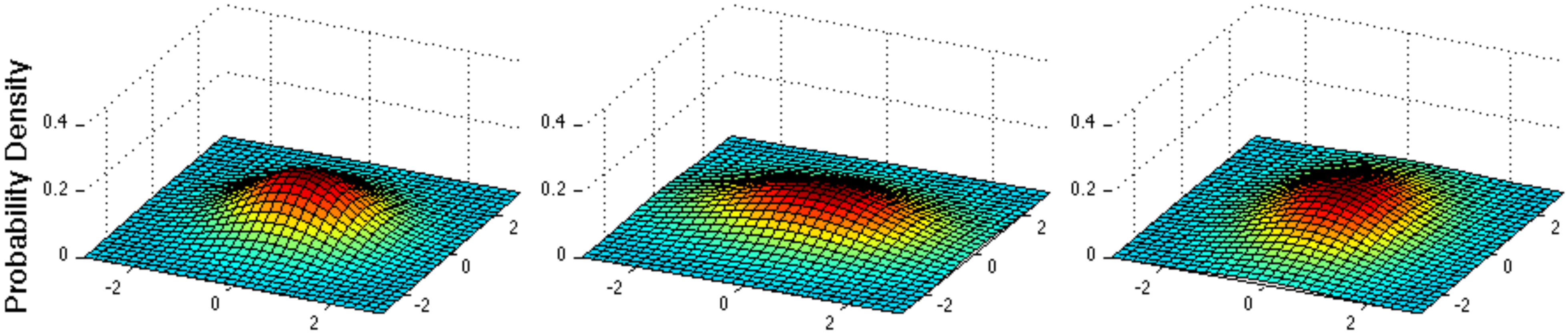
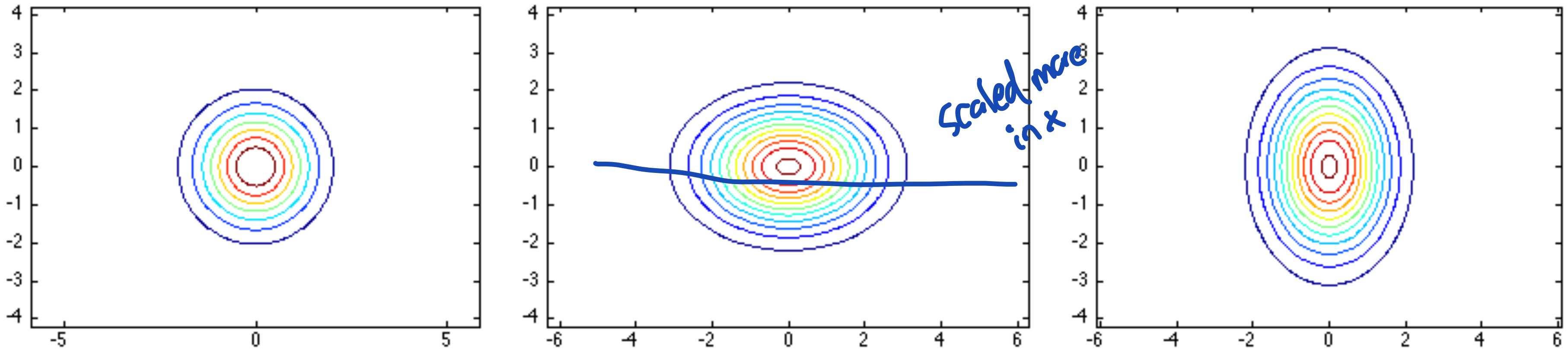


Figure: Probability density function



scaled more in x

Figure: Contour plot of the pdf

Bivariate Gaussian

$$\boldsymbol{\Sigma} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \qquad \boldsymbol{\Sigma} = \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix} \qquad \boldsymbol{\Sigma} = \begin{pmatrix} 1 & 0.8 \\ 0.8 & 1 \end{pmatrix}$$

$$= \mathbf{Q}_1 \begin{pmatrix} 1.5 & 0. \\ 0. & 0.5 \end{pmatrix} \mathbf{Q}_1^\top \qquad = \mathbf{Q}_2 \begin{pmatrix} 1.8 & 0. \\ 0. & 0.2 \end{pmatrix} \mathbf{Q}_2^\top$$

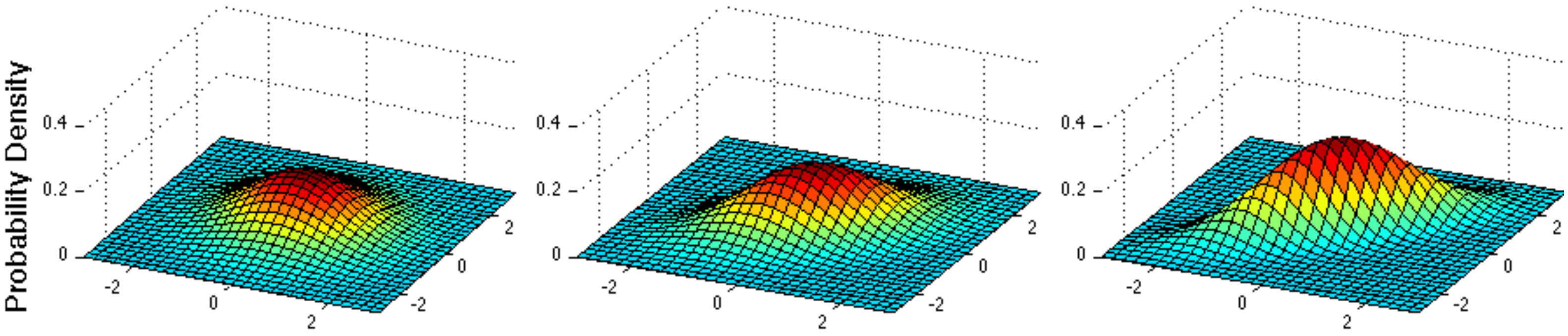


Figure: Probability density function

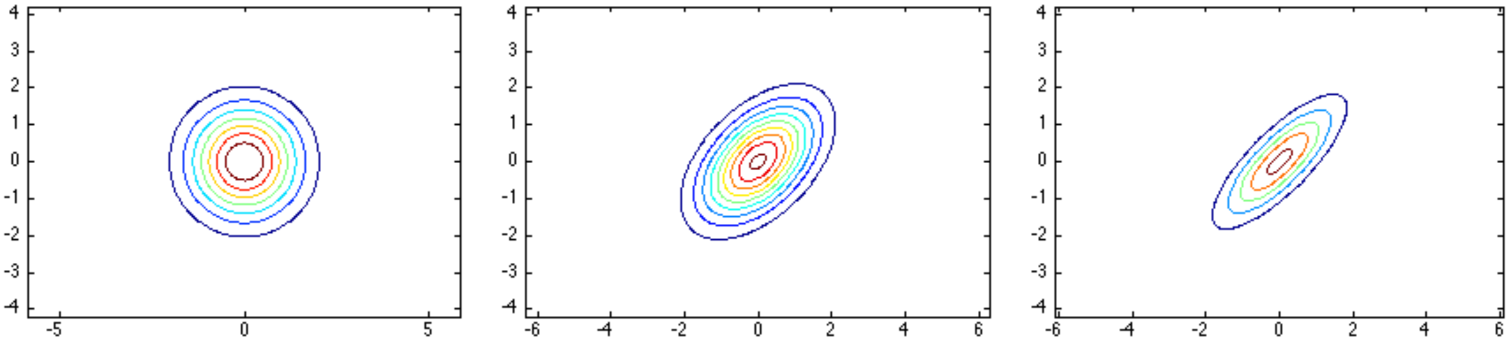


Figure: Contour plot of the pdf

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

Maximum Likelihood for Multi~~variate~~ Gaussian

model the temperature highs

e.g. -2.5, -9.9, -12.1, ...

and find mean of Gaussian

1. Write down likelihood
2. maximize likelihood

$$\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

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.

temperatures: $x_1, x_2, \ldots x_n$

i.i.d

$$L(\underset{\mu,\sigma}{\underset{\uparrow}{\theta}}) = \prod_{i=1}^{N} \underbrace{\frac{1}{\sqrt{2\pi\sigma^2}}}_{\text{term goes away}} e^{-\frac{(x_i-\mu)^2}{2\sigma^2}}$$

$$\theta^{N_H}(1-\theta)^{N_T}$$

$$\underset{\mu}{\operatorname{argmax}}\, L(\mu) = \underset{\mu}{\operatorname{argmax}}\, \ell(\mu)$$

$$= \underset{\mu}{\operatorname{argmax}} \sum_{i=1}^{N} \log\left(e^{-\frac{(x_i-\mu)^2}{2\sigma^2}}\right)$$

$$= \underset{\mu}{\operatorname{argmax}} \sum_{i=1}^{N} -\frac{(x_i-\mu)^2}{2\sigma^2} \Big\} \text{positive constant}$$

$$= \underset{\mu}{\operatorname{argmax}} \sum_{i=1}^{N} -(x_i-\mu)^2$$

$$= \underset{\mu}{\operatorname{argmin}} \sum_{i=1}^{N} (x_i-\mu)^2$$

$$\frac{d}{d\mu}\ell(\mu) = 2\sum_{i=1}^{N}(x_i-\mu) = 0$$

$$\sum_{i=1}^{N} x_i = N\mu$$

$$\mu = \frac{1}{N} \sum_{i=1}^{N} x_i$$

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

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

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:

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

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 $\mathbf{\Sigma}$.

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

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

The best estimate for $\mathbf{\Sigma}$ is the empirical covariance.

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:

2 ways

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

- 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.

$D = \{x_i, y_i\}_{i=1}^{N}$

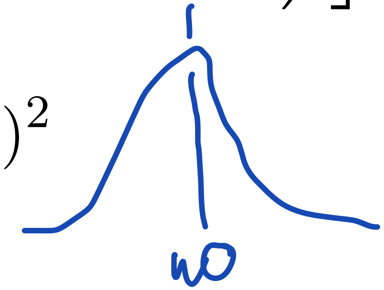
$$\underbrace{t \,|\, \mathbf{x} \sim \mathcal{N}(\mathbf{w}^\top \mathbf{x},\ \sigma^2)}_{}$$

- Linear regression is just maximum likelihood under this model:

$$\frac{1}{N}\sum_{i=1}^{N} \log p(t^{(i)} \,|\, \mathbf{x}^{(i)}; \mathbf{w}, b) = \frac{1}{N}\sum_{i=1}^{N} \log \mathcal{N}(t^{(i)}; \mathbf{w}^\top \mathbf{x}, \sigma^2)$$

$$= \frac{1}{N}\sum_{i=1}^{N} \log\left[\frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(t^{(i)} - \mathbf{w}^\top \mathbf{x})^2}{2\sigma^2}\right)\right]$$

$$= \text{const} - \frac{1}{2N\sigma^2}\sum_{i=1}^{N}(t^{(i)} - \mathbf{w}^\top \mathbf{x})^2$$



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

$$D = \{(x_1, t_1)\ (x_2, t_2) \ldots (x_N, t_N)\}$$

$$L(D \mid \underset{\underset{w}{\nearrow}}{\theta}) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \, e_x \frac{(t_i - w^T x_i)^2}{2\sigma^2} \cdot \underline{\underline{e^{-c|w|_2^2}}}$$

how likely is t_i given $N(w^T x_i, \sigma^2)$?

$$\underset{w}{\text{argmax}}\ \ell(D \mid w) = \underset{w}{\text{argmax}} \sum - \frac{(t_i - w^T x_i)^2}{2\sigma^2} - c|w|^2$$

$$= \underset{w}{\text{argmin}}\ \frac{1}{2\sigma^2} \sum (\underset{\text{target}}{t_i} - \underset{\text{prediction}}{w^T x_i})^2 + c|w|^2$$

$$\underbrace{\overbrace{L(D \mid \theta)}^{\text{MLE}}\ \underset{\underset{\text{prior}}{\uparrow}}{L(\theta)}}_{\text{MAP}}$$

maximizing likelihood under Gaussian model for data is equivalent to minimizing LS

″ max posterior => adding regularization to least squares

3 lectures left

Today : Gaussian Discriminant Analysis + PCA
matrix factorization

next week : ethics module

final week : k-means, EM algorithm / RL

Final: April 14th

Regularization as MAP Inference

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

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

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

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

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

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

- 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!

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?

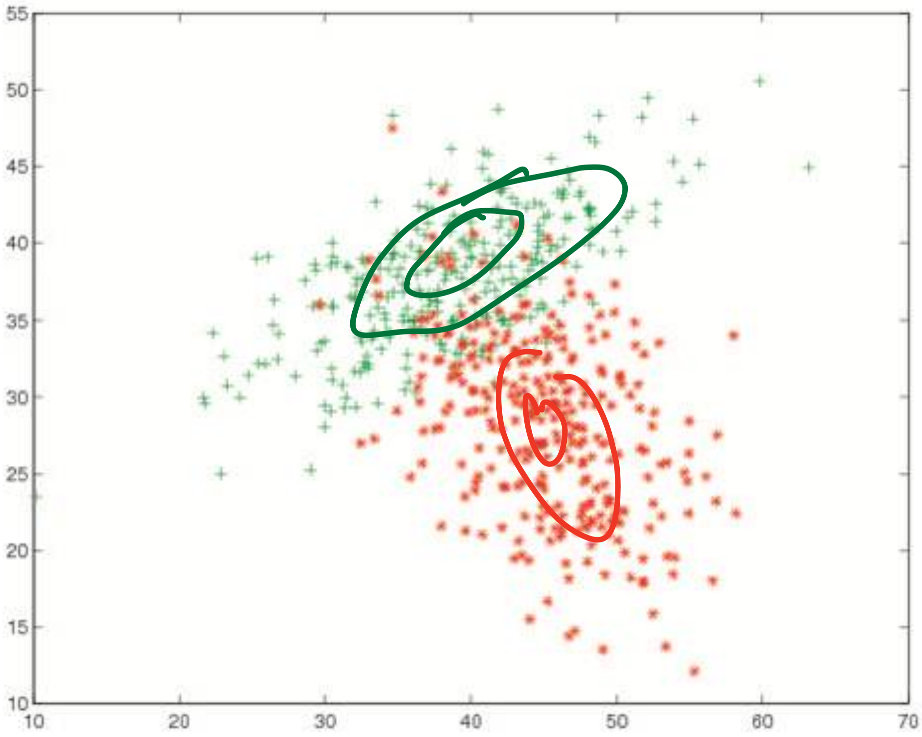
$p(x)$

binary classification

$\max_{t} p(t|x)$

Classification: Diabetes Example

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



- $p(\mathbf{x} \mid t = k)$ for each class is shaped like an ellipse

$\Longrightarrow$ we model each class as a multivariate Gaussian

Gaussian Discriminant Analysis

$x^T A x$

$\mathbb{R}^D \to \mathbb{R}$

$\overline{2\sigma^2}$

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

normalizing constant

$\Rightarrow \mathbb{R}$ quadratic form

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

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

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

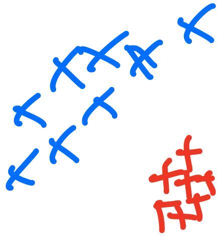
GDA: Learning

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

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

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

spam



$$\begin{aligned}
\phi_1 &= \frac{1}{N}\sum_{i=1}^{N} r_1^{(i)} \\
\boldsymbol{\mu}_k &= \frac{\sum_{i=1}^{N} r_k^{(i)} \cdot \mathbf{x}^{(i)}}{\sum_{i=1}^{N} r_k^{(i)}} \\
\boldsymbol{\Sigma}_k &= \frac{1}{\sum_{i=1}^{N} r_k^{(i)}} \sum_{i=1}^{N} r_k^{(i)} (\mathbf{x}^{(i)} - \boldsymbol{\mu}_k)(\mathbf{x}^{(i)} - \boldsymbol{\mu}_k)^\top \\
\underline{\underline{r_k^{(i)}}} &= \mathbb{1}[t^{(i)} = k]
\end{aligned}$$

prior probability $\frac{\#\text{ spam}}{\#\text{ total}}$

(average feature vectors for all spam emails)

(sum of outer products)

GDA Decision Boundary

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

prior prob. ← multivariate gaussian

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

(denominator same for every class)

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

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

$\log p(t_1|x) = \log p(t_2|x)$

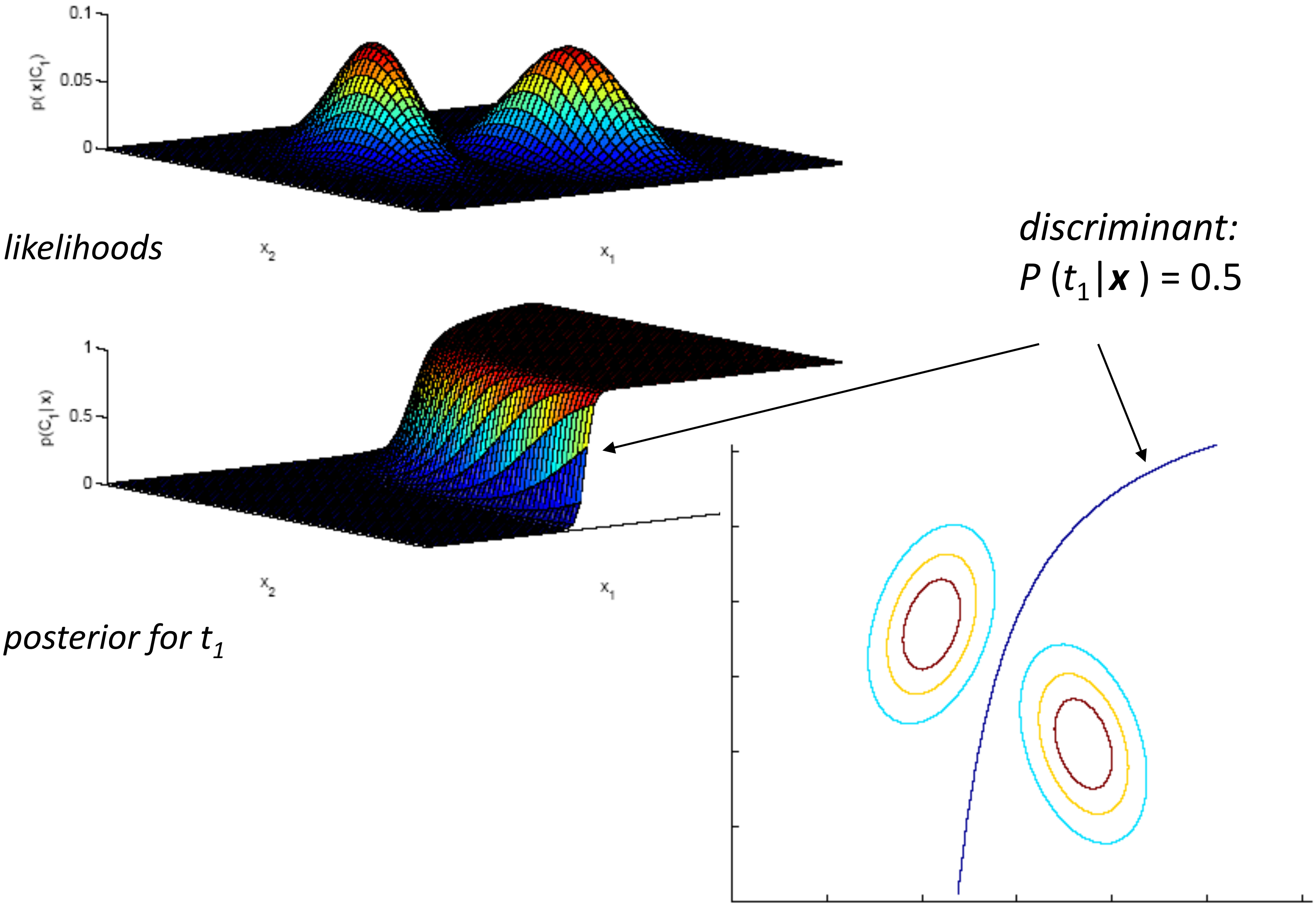
- Decision boundary:

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

quadratic terms of x

- 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



likelihoods

posterior for t_1

discriminant:
$P(t_1|\boldsymbol{x}) = 0.5$

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. $\mathbf{x}$):

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

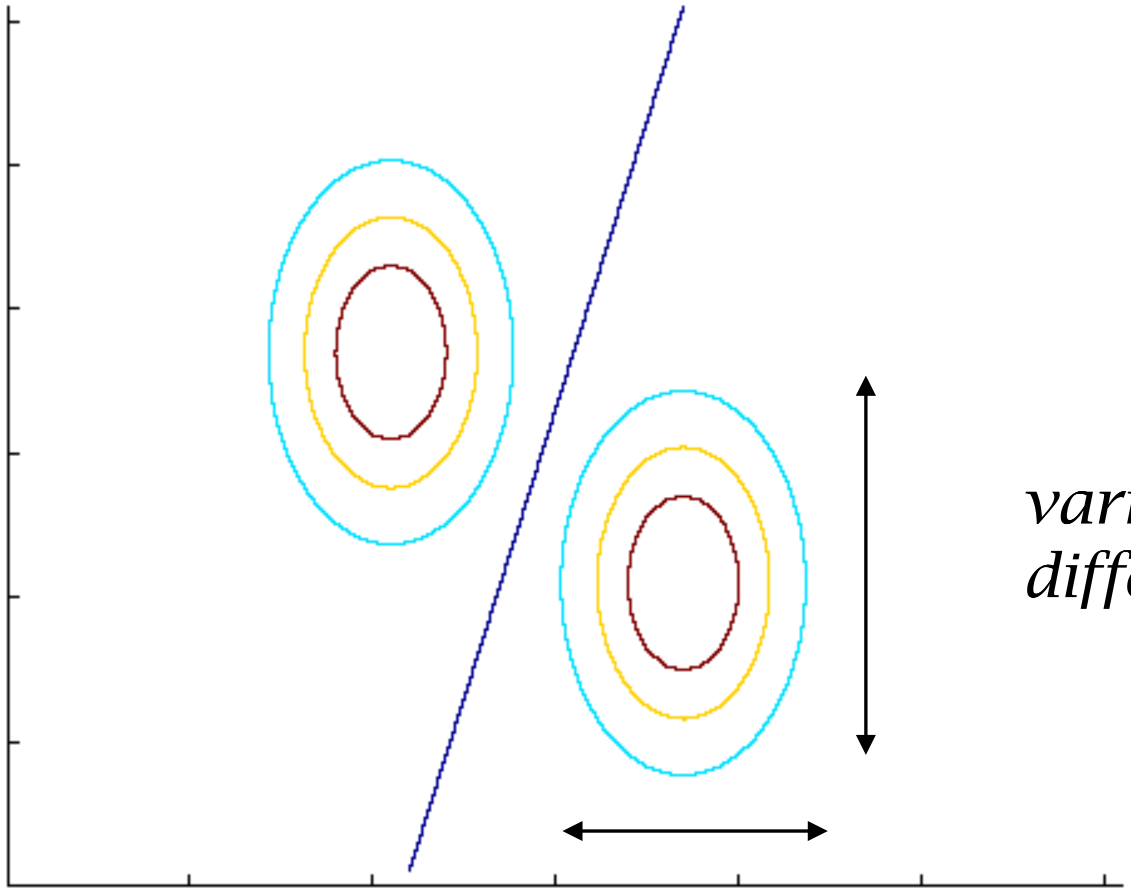
- What if all classes share the same covariance $\boldsymbol{\Sigma}$?
  - We get a linear decision boundary!

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

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

linear function of x

GDA Decision Boundary: Shared Covariances



variances may be different

GDA vs Logistic Regression

- Binary classification: If you examine $p(t = 1 \,|\, \mathbf{x})$ under GDA and assume $\boldsymbol{\Sigma}_0 = \boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}$, 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 $(\phi, \boldsymbol{\mu}_0, \boldsymbol{\mu}_1, \boldsymbol{\Sigma})$.

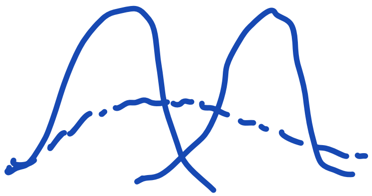
- Same model as logistic regression!

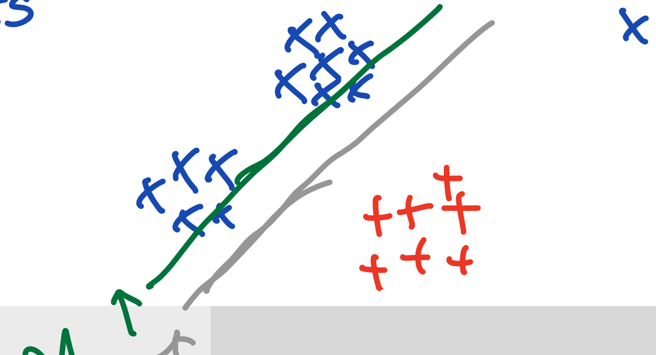
GDA vs Logistic Regression

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?)

→ ignore features with missing datapoints





Gaussian Naive Bayes

GDA

(potentially hard to estimate mean)

logistic regression

When the data is non-Gaussian

- What if $\mathbf{x}$ is high-dimensional?
  - The $\mathbf{\Sigma}_k$ 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. $\mathbf{\Sigma}$ is diagonal.
  - Hence, only D parameters for $\mathbf{\Sigma}$!

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

$$\begin{aligned} \mu_{jk} &= \frac{\sum_{i=1}^{N} r_k^{(i)} \, x_j^{(i)}}{\sum_{i=1}^{N} r_k^{(i)}} \\ \sigma_{jk}^2 &= \frac{\sum_{i=1}^{N} r_k^{(i)} \, (x_j^{(i)} - \mu_{jk})^2}{\sum_{i=1}^{N} r_k^{(i)}} \\ r_k^{(i)} &= \mathbb{1}[t^{(i)} = k] \end{aligned}$$

Decision Boundary: Isotropic

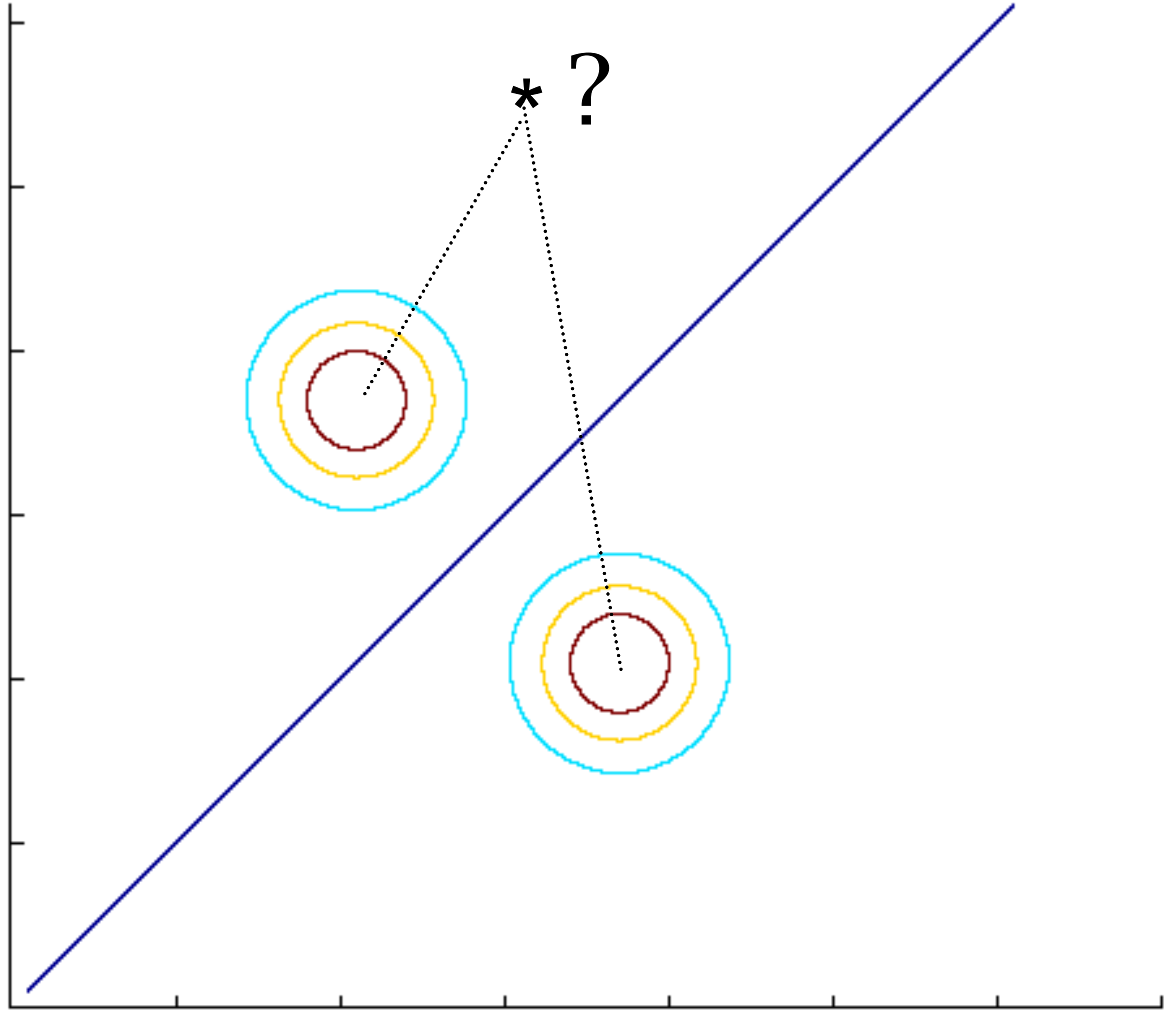
- We can go even further and assume the covariances are spherical, or isotropic.
- In this case: $\mathbf{\Sigma} = \sigma^2 \mathbf{I}$ (just need one parameter!)
- Going back to the class posterior for GDA:

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

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

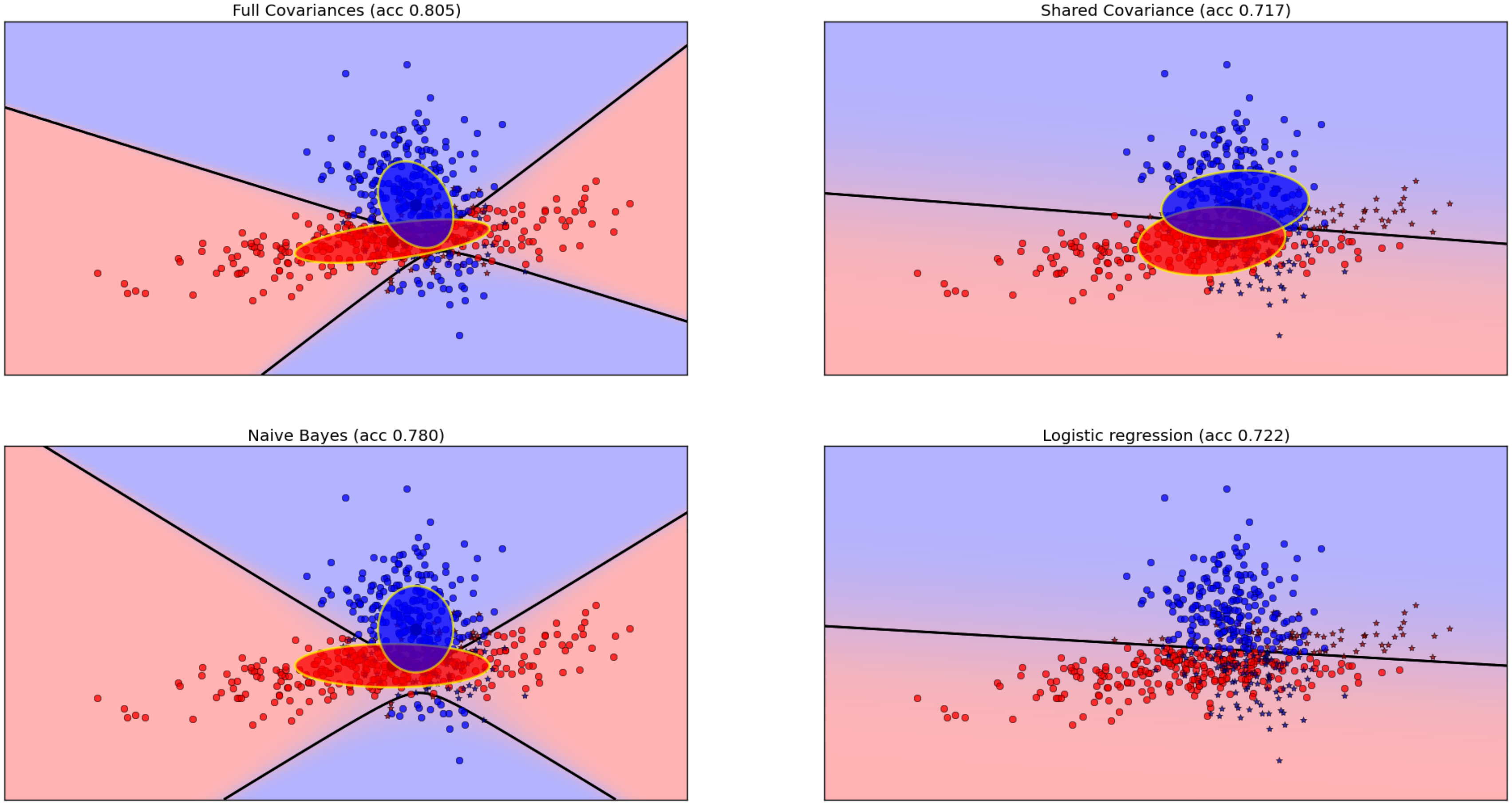
$$\begin{aligned}\log p(t_k \,|\, \mathbf{x}) - \log p(t_\ell \,|\, \mathbf{x}) &= -\frac{1}{2\sigma^2}\left[(\mathbf{x} - \boldsymbol{\mu}_k)^\top(\mathbf{x} - \boldsymbol{\mu}_k) - (\mathbf{x} - \boldsymbol{\mu}_\ell)^\top(\mathbf{x} - \boldsymbol{\mu}_\ell)\right] \\ &= -\frac{1}{2\sigma^2}\left[\|\mathbf{x}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

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.

linear decision boundary (less likely to overfit)

- 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.

7 3 — — 4 ....

(skip the ones that are missing)

- Tries to solve a hard problem (model $p(\mathbf{x})$) in order to solve a easy problem (model $p(t \mid \mathbf{x})$).

Next up: Unsupervised learning with PCA!

GPT

discriminative or generative?

$w_1, w_2, \ldots\ldots w_n$

$\max p(w_n \mid w_1, w_2, \ldots w_{n-1})$

generative model of distribution of words