CSC 311: Introduction to Machine Learning

Lecture 8 - Multivariate Gaussians, GDA

Rahul G. Krishnan   Alice Gao

University of Toronto, Fall 2022

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.

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 vector $\mathbf{v}$ such that

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

  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.

$$A = \begin{pmatrix} 1 & 3 \\ 3 & 9 \end{pmatrix}$$

$$\lambda_1 = 10, \quad v_1 = \begin{pmatrix} 1 \\ 3 \end{pmatrix} \qquad \underset{A}{\begin{pmatrix} 1 & 3 \\ 3 & 9 \end{pmatrix}} \underset{v_1}{\begin{pmatrix} 1 \\ 3 \end{pmatrix}} = \underset{\lambda_1}{10} \underset{v_1}{\begin{pmatrix} 1 \\ 3 \end{pmatrix}}$$

$$\lambda_2 = 0, \quad v_2 = \begin{pmatrix} -3 \\ 1 \end{pmatrix} \qquad \underset{A}{\begin{pmatrix} 1 & 3 \\ 3 & 9 \end{pmatrix}} \underset{v_2}{\begin{pmatrix} -3 \\ 1 \end{pmatrix}} = \underset{\lambda_2}{0} \underset{v_2}{\begin{pmatrix} -3 \\ 1 \end{pmatrix}}$$

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.

perpendicular unit vectors.

Spectral Decomposition

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

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

where

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

Check that this is reasonable:

$$\mathbf{A}\mathbf{q}_i = \lambda_i q_i \implies AQ = Q\Lambda \implies A = Q\Lambda Q^{-1}.$$

$$A(q_1\ q_2 \cdots q_N) = (q_1\ q_2 \cdots q_N)\begin{pmatrix}\lambda_1 & & 0\\ & \lambda_2 & \\ & & \ddots \\ 0 & & & \lambda_N\end{pmatrix}$$

$$A = \begin{pmatrix} 1 & 3 \\ 3 & 9 \end{pmatrix} \qquad Q = \frac{1}{\sqrt{10}} \begin{pmatrix} 1 & -3 \\ 3 & 1 \end{pmatrix} \qquad \Lambda = \begin{pmatrix} 10 & 0 \\ 0 & 0 \end{pmatrix}$$

$$Q \Lambda Q^T = \frac{1}{\sqrt{10}} \begin{pmatrix} 1 & -3 \\ 3 & 1 \end{pmatrix} \begin{pmatrix} 10 & 0 \\ 0 & 0 \end{pmatrix} \frac{1}{\sqrt{10}} \begin{pmatrix} 1 & 3 \\ -3 & 1 \end{pmatrix}$$

$$= \frac{1}{10} \begin{pmatrix} 10 & 0 \\ 30 & 0 \end{pmatrix} \begin{pmatrix} 1 & 3 \\ -3 & 1 \end{pmatrix}$$

$$= \frac{1}{10} \begin{pmatrix} 10 & 30 \\ 30 & 90 \end{pmatrix} = \begin{pmatrix} 1 & 3 \\ 3 & 9 \end{pmatrix}$$

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:

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

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

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

$$x = \begin{pmatrix} 1 \\ 2 \end{pmatrix} \qquad Q = \frac{1}{\sqrt{10}} \begin{pmatrix} 1 & -3 \\ 3 & 1 \end{pmatrix}$$

$$\tilde{x} = Q^T x = \frac{1}{\sqrt{10}} \begin{pmatrix} 1 & 3 \\ -3 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 2 \end{pmatrix} = \frac{1}{\sqrt{10}} \begin{pmatrix} 7 \\ -1 \end{pmatrix}$$

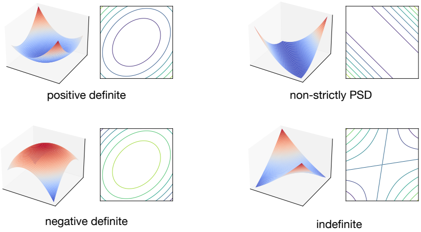
$$x = Q \hat{x} = \frac{1}{\sqrt{10}} \begin{pmatrix} 1 & -3 \\ 3 & 1 \end{pmatrix} \frac{1}{\sqrt{10}} \begin{pmatrix} 7 \\ -1 \end{pmatrix} = \frac{1}{10} \begin{pmatrix} 10 \\ 20 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

$$x = \begin{pmatrix} 1 \\ 2 \end{pmatrix} = \underbrace{\frac{7}{\sqrt{10}}}_{\tilde{x}_1} \Bigg(\underbrace{\frac{1}{\sqrt{10}} \begin{pmatrix} 1 \\ 3 \end{pmatrix}}_{q_1} \Bigg) + \underbrace{\frac{-1}{\sqrt{10}}}_{\tilde{x}_2} \Bigg(\underbrace{\frac{1}{\sqrt{10}} \begin{pmatrix} -3 \\ 1 \end{pmatrix}}_{q_2} \Bigg)$$

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 then $\mathbf{A}$ is indefinite.



$$A = \begin{pmatrix} 1 & 3 \\ 3 & 9 \end{pmatrix} \qquad v = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}$$

$$v^T A v = \begin{pmatrix} v_1 & v_2 \end{pmatrix} \begin{pmatrix} 1 & 3 \\ 3 & 9 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}$$

$$= \begin{pmatrix} v_1 + 3v_2 & 3v_1 + 9v_2 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}$$

$$= (v_1 + 3v_2) v_1 + (3v_1 + 9v_2) v_2$$

$$= v_1^2 + 3v_1 v_2 + 3v_1 v_2 + 9v_2^2$$

$$= v_1^2 + 6v_1 v_2 + 9v_2^2$$

PSD Matrices

- **Exercise:** Show from the definition that nonnegative linear combinations of PSD matrices are PSD.

Assume A_i is PSD, $v^T A_i v \geq 0$ for all v, $a_1, \dots, a_n \geq 0$

$$v^T \left(\sum_i a_i A_i \right) v = \sum_i a_i \left(v^T A_i v \right) \geq 0$$

- **Related:** If $\mathbf{A}$ is a random matrix which is always PSD, then $\mathbb{E}[\mathbf{A}]$ is PSD. (The discrete case is a special case of the above.)
- **Exercise:** Show that for any matrix $\mathbf{B}$, the matrix $\mathbf{B}\mathbf{B}^\top$ is PSD.

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

- **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.)

- **Claim:** $\mathbf{A}$ is positive definite iff all of its eigenvalues are positive. It is PSD iff all of its eigenvalues are nonnegative.
  - Expressing $\mathbf{v}$ in terms of the eigenbasis, $\tilde{\mathbf{v}} = \mathbf{Q}^\top \mathbf{v}$,

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

$$(v_1\ v_2 \cdots v_N) \begin{pmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_N \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_N \end{pmatrix}$$

change of coordinate system.

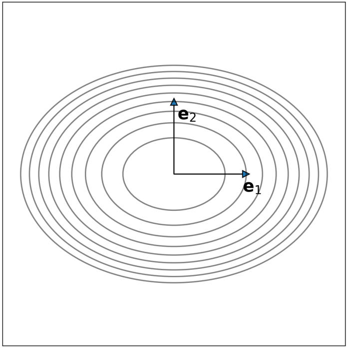
  - This is positive (nonnegative) for all $\mathbf{v}$ iff all the λ_i are positive (nonnegative).

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.

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

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

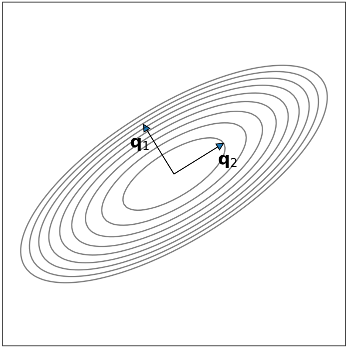


PSD Matrices

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

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

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



- In this example, $\lambda_1 > \lambda_2$.
- All symmetric matrices are diagonal if you choose the right coordinate system.

Matrix Powers

- Applying the Spectral Decomposition, we can square a symmetric matrix:

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

- We can take the k-th power of the matrix:

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

- If $\mathbf{A}$ 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 $\mathbf{A}$ is PSD, then we can easily define the matrix square root:

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

- Observe that $\mathbf{A}^{1/2}$ is PSD and $(\mathbf{A}^{1/2})^2 = \mathbf{A}$.
  This is the unique PSD matrix with this property.

Determinant Properties

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

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

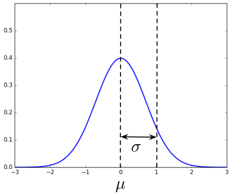
Corollary: the determinant of a positive semi-definite matrix is non-negative, and the determinant of a positive definite matrix is positive.

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 $|\boldsymbol{\Lambda}| = \prod_i \lambda_i$.

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 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

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

Covariance

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

how different dimensions interact.

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

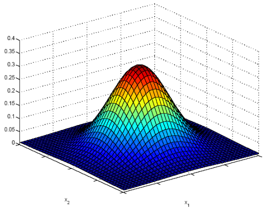
PDF of Gaussian Distribution

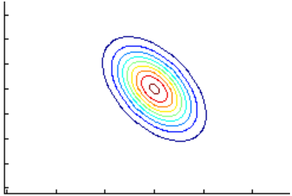
PDF of the univariate Gaussian distribution $(d = 1,\ \boldsymbol{\Sigma} = \sigma^2)$:

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

PDF of the multivariate Gaussian distribution:

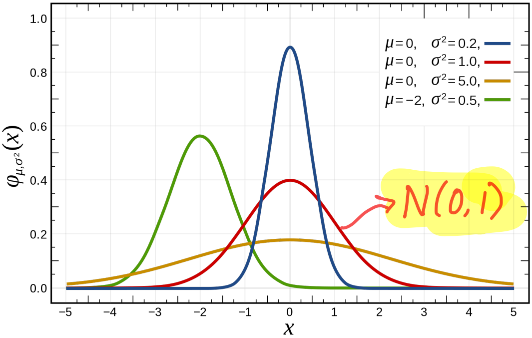
$$\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right]$$





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



Multivariate Shift + Scale

- Any multivariate Gaussian distribution is a shifted and "scaled" version of the standard multivariate normal distribution.
  - The standard multivariate normal has $\boldsymbol{\mu} = \mathbf{0}$ and $\boldsymbol{\Sigma} = \mathbf{I}$
- Multivariate analog of the shift is simple: it's a vector $\boldsymbol{\mu}$
- But what about the scale?
  - In the univariate case, the scale factor was the square root of the variance: $\sigma = \sqrt{\sigma^2}$
  - But in the multivariate case, the covariance $\boldsymbol{\Sigma}$ is a matrix! Does $\boldsymbol{\Sigma}^{\frac{1}{2}}$ exist, and can we scale by it?

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

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

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

$S = \Sigma^{1/2}$

$b = \mu.$

- 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

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

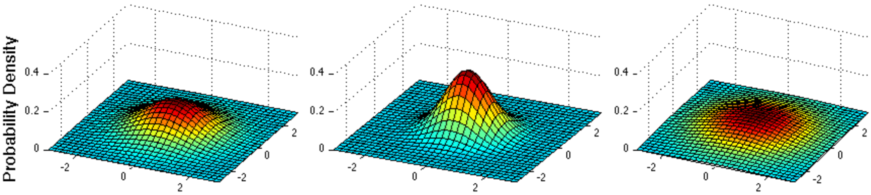


Figure: Probability density function

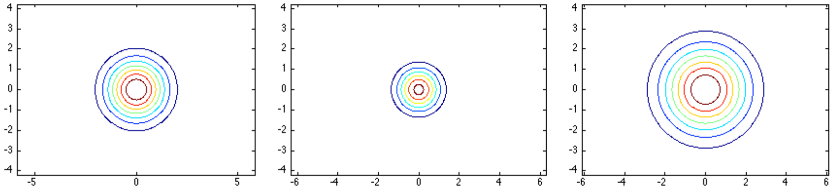


Figure: Contour plot of the pdf

Bivariate Gaussian

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

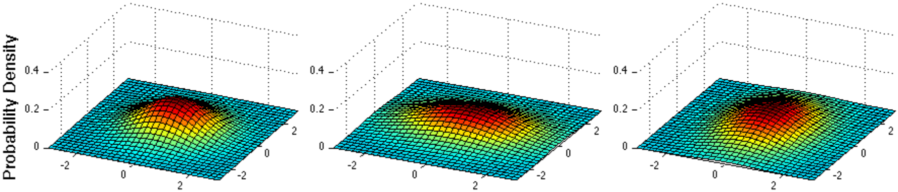


Figure: Probability density function

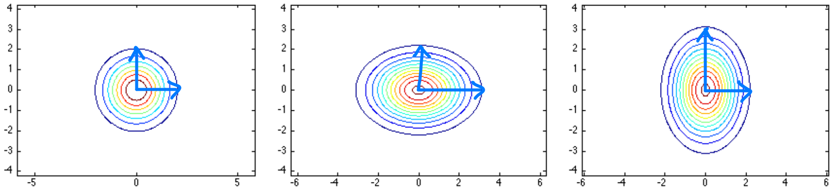


Figure: Contour plot of the pdf

Bivariate Gaussian

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

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

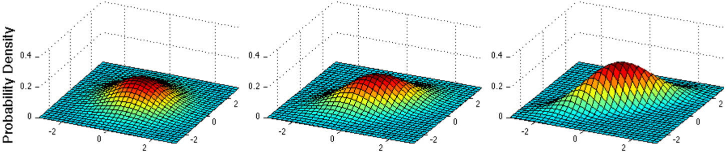


Figure: Probability density function

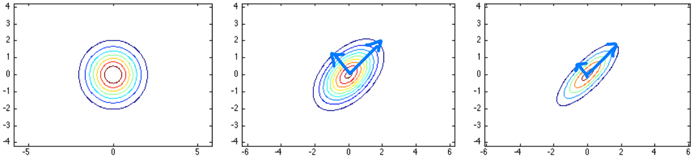


Figure: Contour plot of the pdf

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

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

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

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

$N(\mu, \sigma^2)$

model parameters : μ, σ^2

observations : $x_1, \ldots, x_N$

$$p(x_1, \ldots, x_N \mid \mu, \sigma^2) = \prod_{i=1}^{N} p(x_i \mid \mu, \sigma^2)$$

$$= \prod_{i=1}^{N} \left(\frac{1}{\sqrt{2\pi}\sigma}\right)\left(\exp\left(-\frac{(x_i-\mu)^2}{2\sigma^2}\right)\right)$$

$$\log p(D \mid \mu, \sigma^2) = \sum_{i=1}^{N} \left(\log\frac{1}{\sqrt{2\pi}\sigma}\right) + \left(-\frac{(x_i-\mu)^2}{2\sigma^2}\right)$$

$$= \sum_{i=1}^{N} -\log\sqrt{2\pi}\sigma - \frac{(x_i-\mu)^2}{2\sigma^2}$$

$$\log P(D|\mu,\sigma^2) = \sum_{i=1}^{N} -\log\sqrt{2\pi}\sigma - \frac{(x_i-\mu)^2}{2\sigma^2}$$

$$\frac{\partial}{\partial\mu}\log P(D|\mu,\sigma^2) = \sum_{i=1}^{N} + \frac{\cancel{2}}{\cancel{2}\sigma^2}(x_i-\mu)$$

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

$$\Rightarrow \sum_{i=1}^{N}(x_i-\mu) = 0 \Rightarrow \sum_{i=1}^{N} x_i = \sum_{i=1}^{N}\mu = N\mu$$

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

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

$$\log P(D|\mu,\sigma^2) = \log \prod_{i=1}^{N}\left(\frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{(x^{(i)}-\mu)^2}{2\sigma^2}\right)\right)$$

$$= \sum_{i=1}^{N}\log\left(\frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{(x^{(i)}-\mu)^2}{2\sigma^2}\right)\right)$$

$$= \sum_{i=1}^{N}\left(-\log(\sqrt{2\pi}\sigma) - \frac{(x^{(i)}-\mu)^2}{2\sigma^2}\right)$$

$$\frac{\partial \log P(D|\mu,\sigma^2)}{\partial\mu} = \frac{\partial}{\partial\mu}\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{N}(x^{(i)}-\mu)^2\right)$$

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

$$\frac{\partial \log p(D \mid \mu, \sigma^2)}{\partial \mu} = \frac{1}{\sigma^2} \sum_{i=1}^{N} (x^{(i)} - \mu) = 0$$

$$\Longrightarrow \sum_{i=1}^{N} x^{(i)} - N\mu = 0$$

$$\Longrightarrow \mu = \frac{1}{N} \sum_{i=1}^{N} x^{(i)}$$

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:

$$\frac{\mathrm{d}\ell}{\mathrm{d}\boldsymbol{\mu}} = -\sum_{i=1}^{N} \frac{\mathrm{d}}{\mathrm{d}\boldsymbol{\mu}} \frac{1}{2} (\mathbf{x}^{(i)} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x}^{(i)} - \boldsymbol{\mu})$$

$$= -\sum_{i=1}^{N} \boldsymbol{\Sigma}^{-1} (\mathbf{x}^{(i)} - \boldsymbol{\mu}) = 0 \qquad \text{using identity } \nabla_{\mathbf{x}} \mathbf{x}^\top \mathbf{A} \mathbf{x} = 2\mathbf{A}\mathbf{x}$$

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 $\mathbf{\Sigma}$.

$$\frac{\partial \ell}{\partial \mathbf{\Sigma}} = 0$$

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

$$= \frac{1}{N}(\mathbf{X} - \mathbf{1}\boldsymbol{\mu}^\top)^\top(\mathbf{X} - \mathbf{1}\boldsymbol{\mu}^\top)$$

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:

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

- 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.

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

- Linear regression is just maximum likelihood under this model:

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

Regularization as MAP Inference

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

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

- 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!

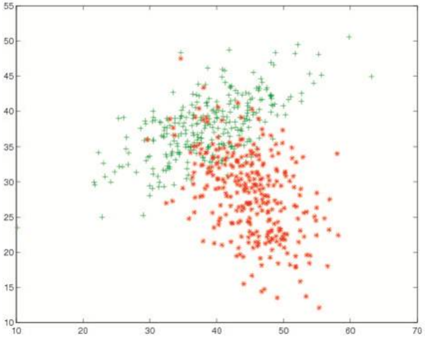
Generative vs Discriminative (Recap)

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?

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

- 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

ϕ: prob. for an example to be in class 1.

- Learn the parameters for each class using maximum likelihood
- For simplicity, assume binary classification $k \in \{1, 2\}$

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

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

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

fraction of examples in class 1.

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

empirical mean.

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

empirical covariance.

$$\hat{r}_k^{(i)} = \mathbb{1}[t^{(i)} = k]$$

does example i belong in class k?

2 classes. $k \in \{1, 2\}$. binary classification.

ϕ is the probability for an example to belong in class 1.

μ_k is the mean vector for class k.

Σ_k is the co-variance matrix for class k.

$$r_k^{(i)} = \mathbb{1}[t^{(i)} = k] \sim \text{does } i^{th} \text{ example belong in class } k\,?$$

$$\hat{\phi} = \frac{1}{N} \sum_{i=1}^{N} r_1^{(i)} \quad \text{fraction of examples in class 1.}$$

$$\hat{\mu}_k = \frac{\sum_{i=1}^{N} r_k^{(i)} \cdot x^{(i)}}{\sum_{i=1}^{N} r_k^{(i)}} \quad \text{empirical mean for examples in class } k.$$

$\hat{\Sigma}_k$ empirical co-variance for examples in class k.

GDA Decision Boundary

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

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

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

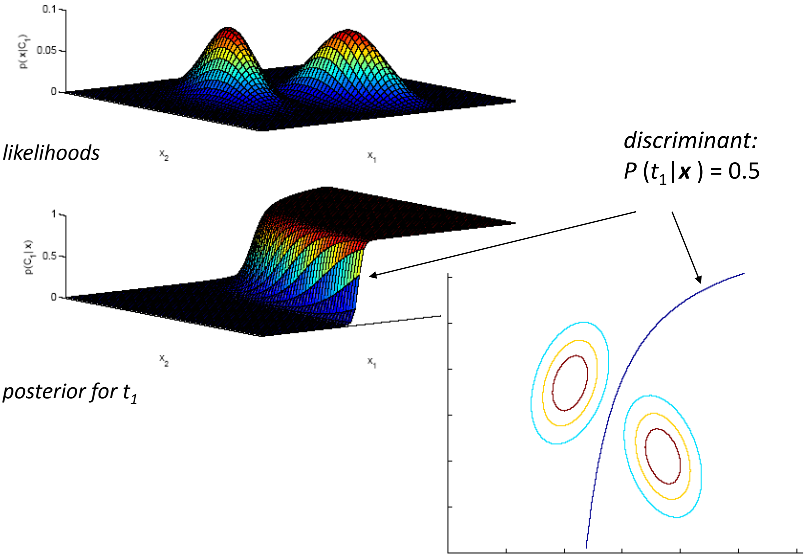
$$\begin{aligned} \log p(t_k|\mathbf{x}) &= \log p(\mathbf{x}|t_k) + \log p(t_k) - \log p(\mathbf{x}) \\ &= -\frac{D}{2}\log(2\pi) - \frac{1}{2}\log|\boldsymbol{\Sigma}_k| - \frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1}(\mathbf{x}-\boldsymbol{\mu}_k) + \\ &\quad + \log p(t_k)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

- 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. $\mathbf{x}$):

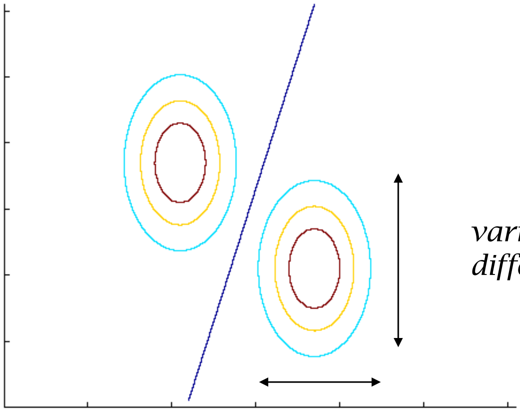
$$\mathbf{x}^\top \boldsymbol{\Sigma}_k^{-1}\mathbf{x} - 2\boldsymbol{\mu}_k^\top \boldsymbol{\Sigma}_k^{-1}\mathbf{x} = \mathbf{x}^\top \boldsymbol{\Sigma}_\ell^{-1}\mathbf{x} - 2\boldsymbol{\mu}_\ell^\top \boldsymbol{\Sigma}_\ell^{-1}\mathbf{x} + \text{Const}$$

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

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

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

GDA Decision Boundary: Shared Covariances



variances may be different

GDA vs Logistic Regression

- Binary classification: If you examine $p(t = 1 \mid \mathbf{x})$ under GDA and assume $\mathbf{\Sigma}_0 = \mathbf{\Sigma}_1 = \mathbf{\Sigma}$, you will find that it looks like this:

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

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

- 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 $\mathbf{\Sigma}_k$ 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

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

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

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

Decision Boundary: Isotropic

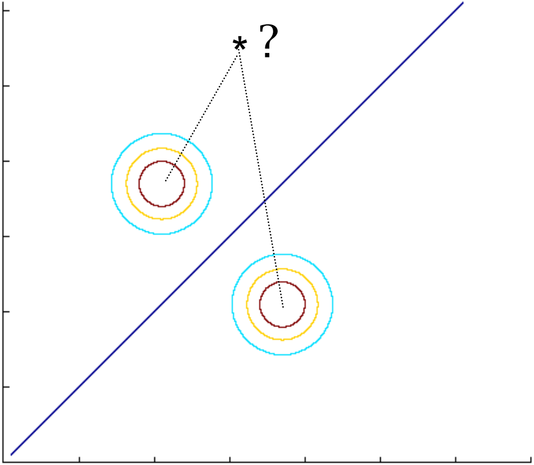
- We can go even further and assume the covariances are spherical, or isotropic.
- In this case: $\mathbf{\Sigma} = \sigma^2 \mathbf{I}$ (just need one parameter!)
- Going back to the class posterior for GDA:

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

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

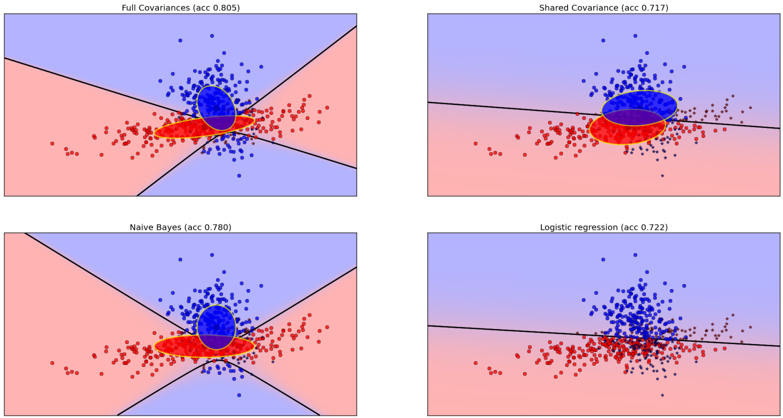
$$\begin{aligned}\log p(t_k \,|\, \mathbf{x}) - \log p(t_\ell \,|\, \mathbf{x}) &= -\frac{1}{2\sigma^2}\left[(\mathbf{x} - \boldsymbol{\mu}_k)^\top(\mathbf{x} - \boldsymbol{\mu}_k) - (\mathbf{x} - \boldsymbol{\mu}_\ell)^\top(\mathbf{x} - \boldsymbol{\mu}_\ell)\right] \\ &= -\frac{1}{2\sigma^2}\left[\|\mathbf{x}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

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.
- 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 $p(t \mid \mathbf{x})$).

Next up: Unsupervised learning with PCA!