CSC 311: Introduction to Machine Learning

Lecture 8 - Multivariate Gaussians, GDA

Roger Grosse Rahul G. Krishnan Guodong Zhang

University of Toronto, Fall 2021

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

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.
- I will have very little to say about the general case, since in this course we will only be concerned with the case of symmetric matrices, which is much simpler.
  - Today's tutorial covers the general case, as well as how to compute eigenvectors/eigenvalues.

Spectral Decomposition

- If a matrix $\mathbf{A}$ is symmetric, then the situation is much simpler, due to a result called the Spectral Theorem.
  - All of the eigenvalues are real-valued.
  - There is a full set of linearly independent eigenvectors (i.e. D for a $D \times D$ matrix).
    - I.e., these eigenvectors form a basis for $\mathbb{R}^D$.
  - These eigenvectors can be chosen to be real-valued.
  - The eigenvectors can be chosen to be orthonormal.
- In this class, we will *only* need to use eigenvectors and eigenvalues in the symmetric case. But it's important to remember why this case is so special.

Spectral Decomposition

- Equivalently to the Spectral Theorem, a symmetric matrix $\mathbf{A}$ can be factorized with the Spectral Decomposition:

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

  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 λ_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 $\mathbb{R}^D$.
- I.e., 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:

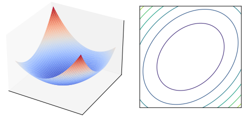
$$\tilde{\mathbf{x}} = \mathbf{Q}^\top \mathbf{x} \qquad \mathbf{x} = \mathbf{Q}\tilde{\mathbf{x}}$$

- In the alternate coordinate system, $\mathbf{A}$ acts by rescaling the individual coordinates (i.e. "stretching" the space):

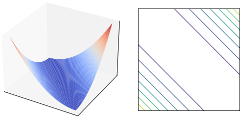
$$\begin{aligned} \mathbf{A}\mathbf{x} &= \tilde{x}_1 \mathbf{A}\mathbf{q}_1 + \cdots + \tilde{x}_D \mathbf{A}\mathbf{q}_D \\ &= \lambda_1 \tilde{x}_1 \mathbf{q}_1 + \cdots + \lambda_D \tilde{x}_D \mathbf{q}_D \end{aligned}$$

PSD Matrices

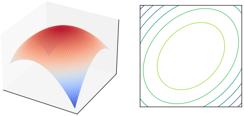
- Symmetric matrices are important because they represent quadratic forms, $f(\mathbf{v}) = \mathbf{v}^\top \mathbf{A} \mathbf{v}$.



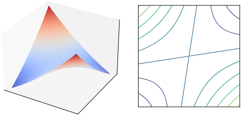
positive definite



non-strictly PSD



negative definite



indefinite

- If $\mathbf{v}^\top \mathbf{A} \mathbf{v} > 0$ for all $\mathbf{v} \neq \mathbf{0}$, i.e. the quadratic form curves upwards, we say that $\mathbf{A}$ is positive definite and denote this $\mathbf{A} \succ \mathbf{0}$.
- If $\mathbf{v}^\top \mathbf{A} \mathbf{v} \geq 0$ for all $\mathbf{v}$, we say $\mathbf{A}$ is positive semidefinite (PSD), denoted $\mathbf{A} \succeq \mathbf{0}$.
- If $\mathbf{v}^\top \mathbf{A} \mathbf{v} < 0$ for all $\mathbf{v} \neq \mathbf{0}$, we say $\mathbf{A}$ 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.

PSD Matrices

- **Exercise:** Show from the definition that nonnegative 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. (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.

- **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.)

PSD Matrices

- **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, $\tilde{\mathbf{v}} = \mathbf{Q}^\top \mathbf{v}$,

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

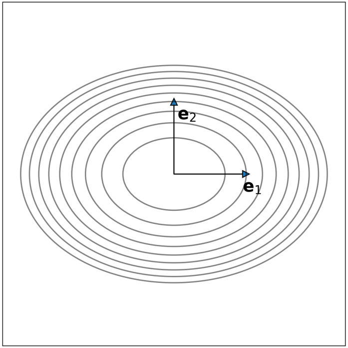
  - This is positive (nonnegative) for all $\mathbf{v}$ iff all the λ_i are positive (nonnegative).

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.

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

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

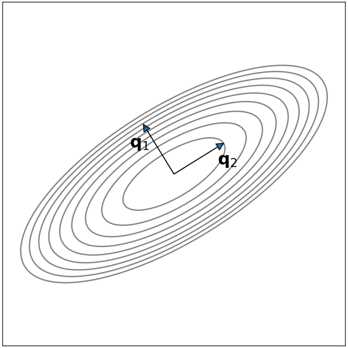


PSD Matrices

- 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

- The Spectral Decomposition makes it easy to compute powers of a matrix. Observe that

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

- Iterating this, for any integer $k > 0$,

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

- Similarly, if $\mathbf{A}$ is invertible, then

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

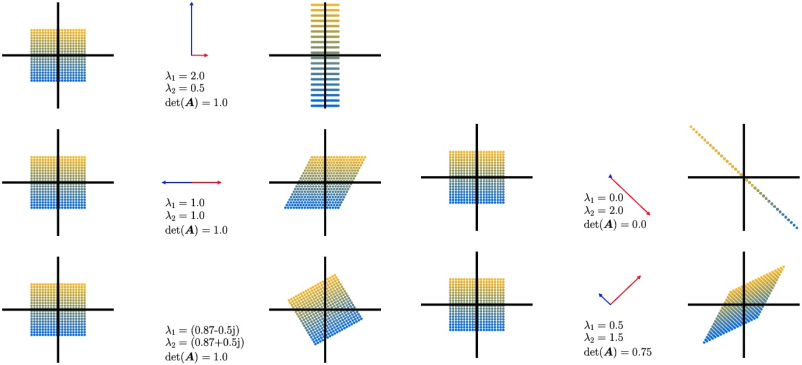
- If $\mathbf{A}$ is PSD, then we can easily define the matrix square root:

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

- 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 (but we won't show this).

Determinant

- The determinant $|\mathbf{B}|$ of a square matrix $\mathbf{B}$ determines how volumes change under linear transformation by $\mathbf{B}$.



- The definition of the determinant is complicated, and we won't need it in this course.

Figure: Mathematics for Machine Learning

Determinant

- Some basic properties:
  - $|\mathbf{BC}| = |\mathbf{B}| \cdot |\mathbf{C}|$
  - $|\mathbf{B}| = 0$ 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 $|\mathbf{\Lambda}| = \prod_i \lambda_i$.
- The determinant of a matrix equals the product of its eigenvalues. This is easy to show in the symmetric case:

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

- **Corollary:** the determinant of a PSD matrix is nonnegative, and the determinant of a positive definite matrix is positive.

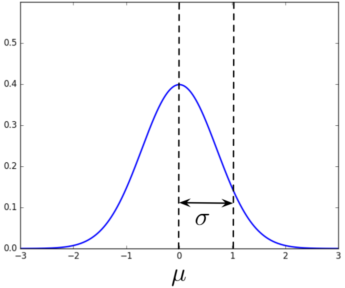
Multivariate Gaussian Distribution

Univariate Gaussian distribution

- Recall the Gaussian, or normal, 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.
- The Central Limit Theorem says that sums of lots of independent random variables are approximately Gaussian.
- In machine learning, we use 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

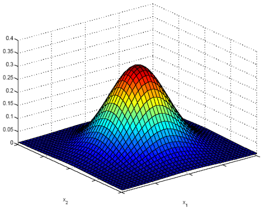
$$\boldsymbol{\Sigma} = \mathrm{Cov}(\mathbf{x}) = \mathbb{E}[(\mathbf{x} - \boldsymbol{\mu})(\mathbf{x} - \boldsymbol{\mu})^\top] = \begin{pmatrix} \sigma_1^2 & \sigma_{12} & \cdots & \sigma_{1D} \\ \sigma_{12} & \sigma_2^2 & \cdots & \sigma_{2D} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{D1} &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

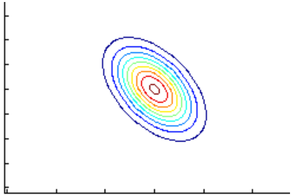
- The statistics ($\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$) uniquely define a multivariate Gaussian (or multivariate Normal) distribution, denoted $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ or $\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$
  - This is not true for distributions in general!

Multivariate Gaussian Distribution

- PDF of the multivariate Gaussian distribution:

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





- Compare to the univariate case ($d = 1$, $\boldsymbol{\Sigma} = \sigma^2$):

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

Bivariate Gaussian

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

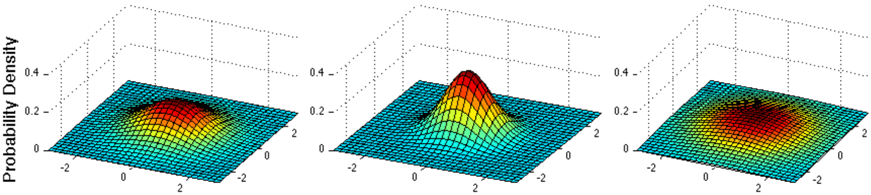


Figure: Probability density function

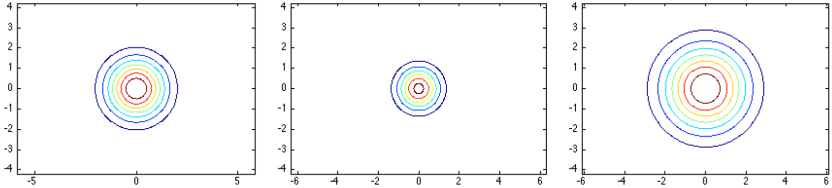


Figure: Contour plot of the pdf

Bivariate Gaussian

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

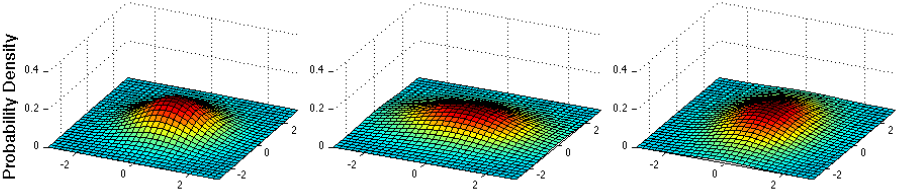


Figure: Probability density function

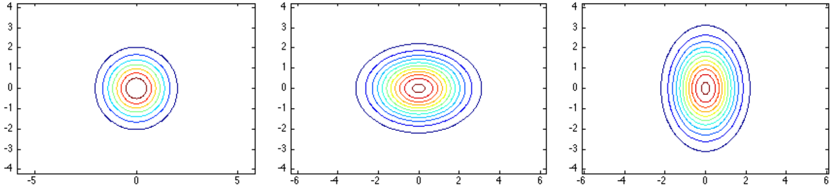


Figure: Contour plot of the pdf

Bivariate Gaussian

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

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

Test your intuition: Does $Q_1 = Q_2$?

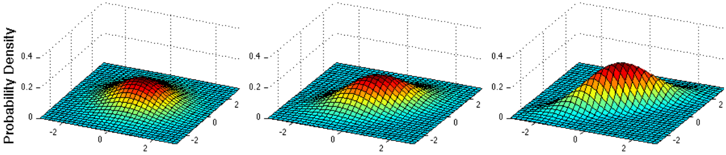


Figure: Probability density function

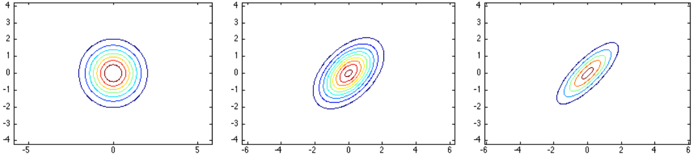


Figure: Contour plot of the pdf

Bivariate Gaussian

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

$$= \mathbf{Q}_1 \begin{pmatrix} 1.5 & 0. \\ 0. & 0.5 \end{pmatrix} \mathbf{Q}_1^\top \qquad = \mathbf{Q}_2 \begin{pmatrix} \lambda_1 & 0. \\ 0. & \lambda_2 \end{pmatrix} \mathbf{Q}_2^\top$$

Test your intuition: Does $Q_1 = Q_2$? What are λ_1 and λ_2?

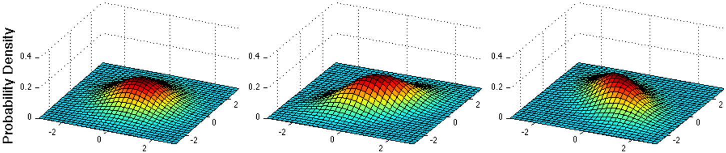


Figure: Probability density function

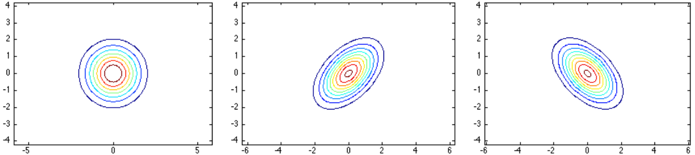
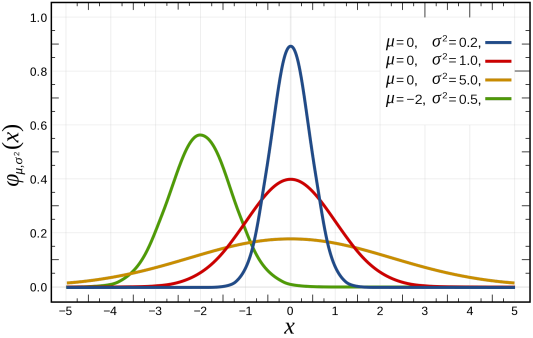


Figure: Contour plot of the pdf

Gaussian Intuition: (Multivariate) Shift + Scale

- Recall that in the univariate case, all normal distributions are shaped like the standard normal distribution
- The densities are related to the standard normal by a shift (μ), a scale (or stretch, or dilation) σ, and a normalization factor



Shift + Scale: Multivariate Case

- Start with a standard (spherical) Gaussian $\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. So $\mathbb{E}[\mathbf{x}] = \mathbf{0}$ and $\mathrm{Cov}(\mathbf{x}) = \mathbf{I}$.
- Consider what happens if we 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,

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

- It's possible to show that $\hat{\mathbf{x}}$ is also Gaussian distributed (but we won't show this here).

Shift + Scale: Multivariate Case

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

- In the univariate case, we obtain $\mathcal{N}(\mu, \sigma^2)$ by starting with $\mathcal{N}(0, 1)$ and shifting by μ and stretching by $\sigma = \sqrt{\sigma^2}$.
- In the multivariate case, to obtain $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, we start with $\mathcal{N}(\mathbf{0}, \mathbf{I})$ and shift by $\boldsymbol{\mu}$ and scale by the matrix square root $\boldsymbol{\Sigma}^{1/2}$.
  - **Recall:** $\boldsymbol{\Sigma}^{1/2} = \mathbf{Q}\boldsymbol{\Lambda}^{1/2}\mathbf{Q}$.
  - **Intuition:** for each eigenvector $\mathbf{q}_i$ with corresponding eigenvalue λ_i, we stretch by a factor of $\sqrt{\lambda_i}$ in the direction $\mathbf{q}_i$.

Gaussian Maximum Likelihood

- Suppose we want to model the distribution of highest and lowest temperatures in Toronto in March, and we've 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 with mean $\boldsymbol{\mu}$, and covariance $\boldsymbol{\Sigma}$. We want to estimate these using data.
- Log-likelihood function:

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

Optional intuition building: why does $|\boldsymbol{\Sigma}|^{1/2}$ show up in the Gaussian density $p(\mathbf{x})$?

Hint: determinant is product of eigenvalues

Gaussian Maximum Likelihood

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

$$0 = \frac{\mathrm{d}\ell}{\mathrm{d}\boldsymbol{\mu}} = -\sum_{i=1}^{N} \frac{\mathrm{d}}{\mathrm{d}\boldsymbol{\mu}} \frac{1}{2} (\mathbf{x}^{(i)} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x}^{(i)} - \boldsymbol{\mu})$$
$$= -\sum_{i=1}^{N} \boldsymbol{\Sigma}^{-1} (\mathbf{x}^{(i)} - \boldsymbol{\mu}) = 0$$

- Here we use the identity $\nabla_{\mathbf{x}} \mathbf{x}^\top \mathbf{A} \mathbf{x} = 2\mathbf{A}\mathbf{x}$
- Solving we get $\hat{\boldsymbol{\mu}} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{x}^{(i)}$. In general, "hat" means estimator
- This is just the sample mean of the observed values, or the empirical mean.

Gaussian Maximum Likelihood

- We can do a similar calculation for the covariance matrix $\boldsymbol{\Sigma}$ (we skip the details).
- Setting the *partial* derivatives to zero, just like before, we get:

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

  where $\mathbf{1}$ is an N-dimensional vector of 1s.
- This is called the empirical covariance and comes up quite often (e.g., PCA soon!)
- Derivation in multivariate case is tedious. No need to worry about it. But it is good practice to derive this in one dimension. See supplement (next slide).

Supplement: MLE for univariate Gaussian

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

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

$$
\hat{\sigma}_{\mathrm{ML}} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (\mathbf{x}^{(i)} - \mu)^2}
$$

Revisiting Linear Regression

Recap: Linear Regression

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

$$y = \mathbf{w}^\top \boldsymbol{\psi}(\mathbf{x})$$

- Squared error loss:

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

- L_2 regularization:

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

- Solution 1: solve analytically by setting the gradient to 0

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

- Solution 2: solve approximately using gradient descent

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

Linear Regression as Maximum Likelihood

- We can give linear regression a probabilistic interpretation by assuming a Gaussian noise model:

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

- Linear regression is just maximum likelihood under this model:

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

Regularization as MAP Inference

- We can view an L_2 regularizer as MAP inference with a Gaussian prior.
- Recall MAP inference:

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

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

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

- 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!

Gaussian Discriminant Analysis

Generative vs Discriminative (Recap)

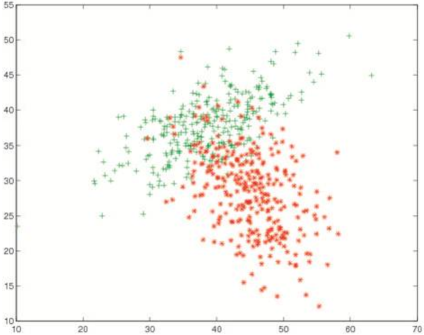
Two approaches to classification:

- 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.



- $p(\mathbf{x} \mid t = k)$ 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 $\mathcal{O}(D^2 K)$ — 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)

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

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 $p(\mathbf{x} \mid t)$:

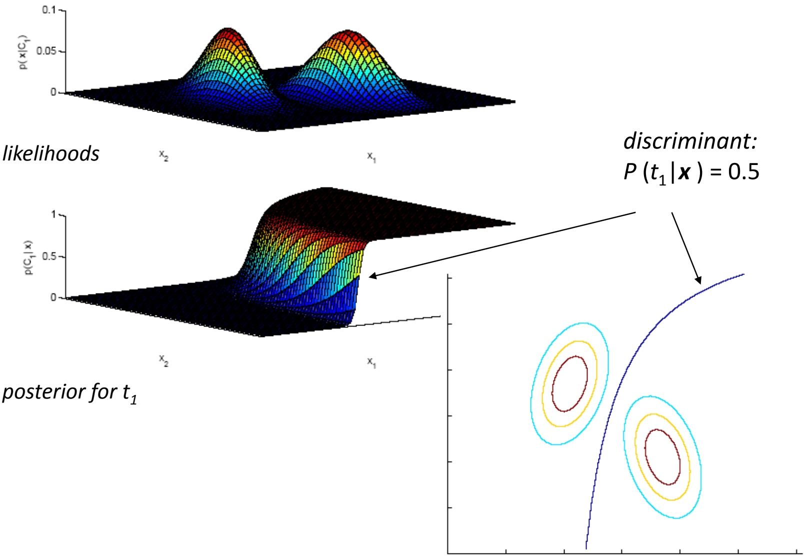
$$\begin{aligned}
\log p(t_k|\mathbf{x}) &= \log p(\mathbf{x}|t_k) + \log p(t_k) - \log p(\mathbf{x}) \\
&= -\frac{D}{2}\log(2\pi) - \frac{1}{2}\log|\boldsymbol{\Sigma}_k| - \frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1}(\mathbf{x} - \boldsymbol{\mu}_k) + \\
&\quad + \log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

- 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:

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

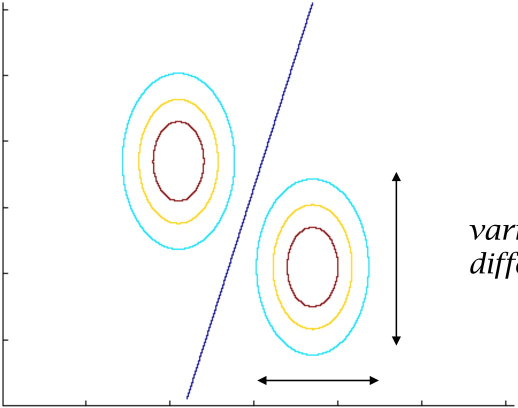
- Expand the product and factor out constants (w.r.t. $\mathbf{x}$):

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

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

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

GDA Decision Boundary: Shared Covariances



variances may be different

GDA 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?)

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

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

- 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:

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

(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

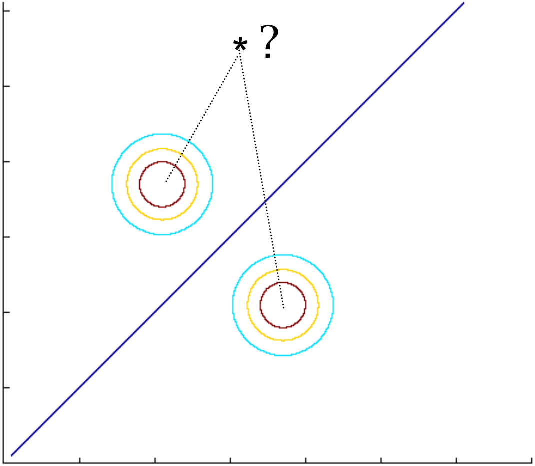
- We can go even further and assume the covariances are spherical, or isotropic.
- In this case: $\boldsymbol{\Sigma} = \sigma^2 \mathbf{I}$ (just need one parameter!)
- Going back to the class posterior for GDA:

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

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

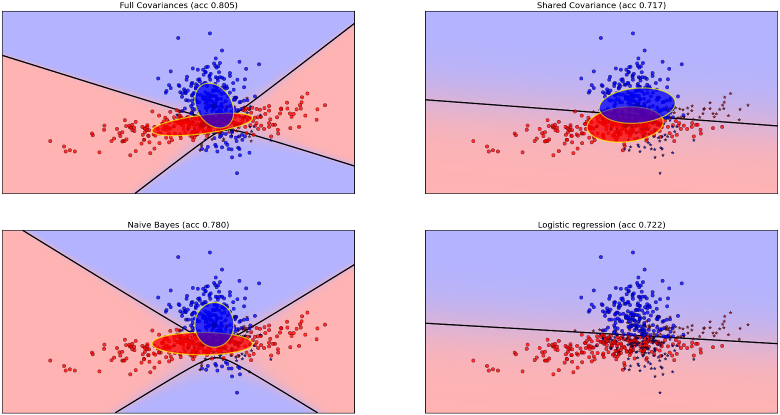
$$\begin{aligned}
\log p(t_k \,|\, \mathbf{x}) - \log p(t_\ell \,|\, \mathbf{x}) &= -\frac{1}{2\sigma^2}\left[(\mathbf{x}-\boldsymbol{\mu}_k)^\top(\mathbf{x}-\boldsymbol{\mu}_k) - (\mathbf{x}-\boldsymbol{\mu}_\ell)^\top(\mathbf{x}-\boldsymbol{\mu}_\ell)\right] \\
&= -\frac{1}{2\sigma^2}\left[\|\mathbf{x}-\boldsymbol{\mu}_k\|^2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

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!