CSC 411 Lecture 09: Generative Models for Classification II

Ethan Fetaya, James Lucas and Emad Andrews

University of Toronto

Today

- Classification - Multi-dimensional (Gaussian) Bayes classifier
- Estimate probability densities from data

Motivation

- Generative models - model $p(\mathbf{x}|t = k)$
- Instead of trying to separate classes, try to model what each class "looks like".
- Recall that $p(\mathbf{x}|t = k)$ may be very complex

$$p(x_1, \cdots, x_d, y) = p(x_1|x_2, \cdots, x_d, y) \cdots p(x_{d-1}|x_d, y)p(x_d, y)$$

- Naive bayes used a conditional independence assumption. What else could we do? Choose a simple distribution.
- Today we will discuss fitting Gaussian distributions to our data.

Bayes Classifier

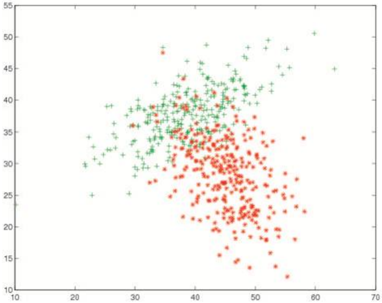
- Let's take a step back...
- Bayes Classifier

$$
\begin{aligned}
h(\mathbf{x}) = \arg\max p(t = k|\mathbf{x}) &= \arg\max \frac{p(\mathbf{x}|t = k)p(t = k)}{p(\mathbf{x})} \\
&= \arg\max p(\mathbf{x}|t = k)p(t = k)
\end{aligned}
$$

- Talked about Discrete **x**, what if **x** is continuous?

Classification: Diabetes Example

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



- How can we model $p(x|t = k)$? Multivariate Gaussian

Gaussian Discriminant Analysis (Gaussian Bayes Classifier)

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

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

  where $|\Sigma_k|$ denotes the determinant of the matrix, and d is dimension of $\mathbf{x}$
- Each class k has associated mean vector $\boldsymbol{\mu}_k$ and covariance matrix Σ_k
- Σ_k has $\mathcal{O}(d^2)$ parameters - could be hard to estimate (more on that later).

Multivariate Data

- Multiple measurements (sensors)
- d inputs/features/attributes
- N instances/observations/examples

$$\mathbf{X} = \begin{bmatrix} x_1^{(1)} & x_2^{(1)} & \cdots & x_d^{(1)} \\ x_1^{(2)} & x_2^{(2)} & \cdots & x_d^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ x_1^{(N)} & x_2^{(N)} & \cdots & x_d^{(N)} \end{bmatrix}$$

Multivariate Parameters

- Mean

$$\mathbb{E}[\mathbf{x}] = [\mu_1, \cdots, \mu_d]^T$$

- Covariance

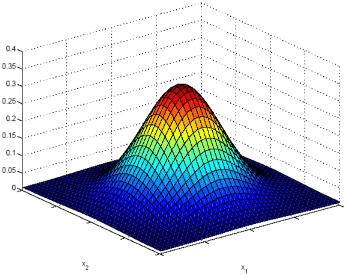
$$\Sigma = Cov(\mathbf{x}) = \mathbb{E}[(\mathbf{x} - \mu)^T(\mathbf{x} - \mu)] = \begin{bmatrix} \sigma_1^2 & \sigma_{12} & \cdots & \sigma_{1d} \\ \sigma_{12} & \sigma_2^2 & \cdots & \sigma_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{d1} & \sigma_{d2} & \cdots & \sigma_d^2 \end{bmatrix}$$

- For Gaussians - all you need to know to represent! (not true in general)

Multivariate Gaussian Distribution

- $\mathbf{x} \sim \mathcal{N}(\mu, \Sigma)$, a Gaussian (or normal) distribution defined as

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



- Mahalanobis distance $(\mathbf{x}-\mu_k)^T\Sigma^{-1}(\mathbf{x}-\mu_k)$ measures the distance from $\mathbf{x}$ to μ in terms of Σ
- It normalizes for difference in variances and correlations

Bivariate Normal

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

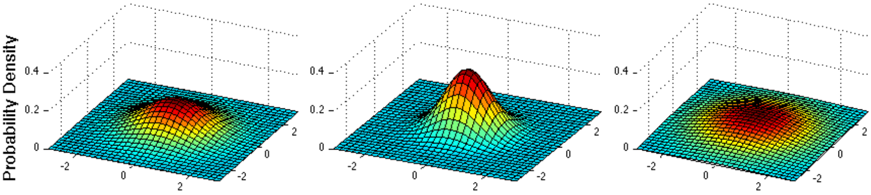


Figure: Probability density function

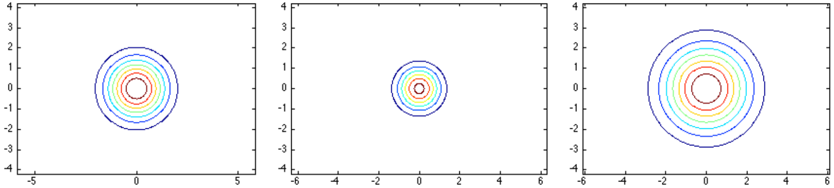


Figure: Contour plot of the pdf

Bivariate Normal

$var(x_1) = var(x_2)$ $\quad$ $var(x_1) > var(x_2)$ $\quad$ $var(x_1) < var(x_2)$

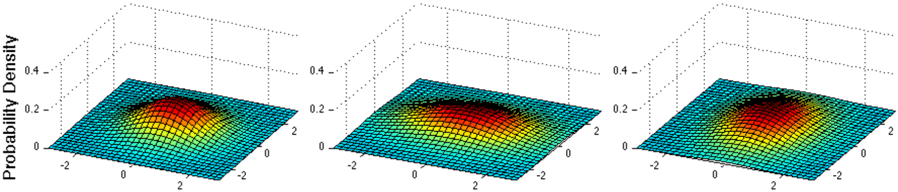


Figure: Probability density function

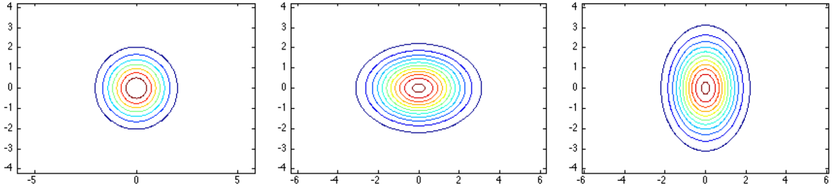


Figure: Contour plot of the pdf

Bivariate Normal

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

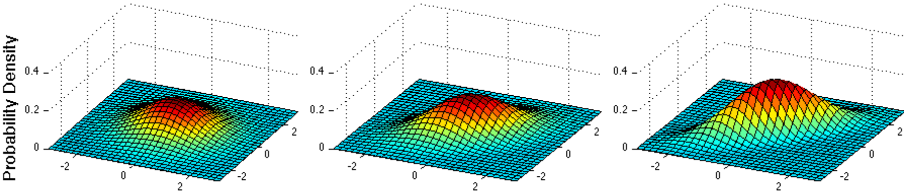


Figure: Probability density function

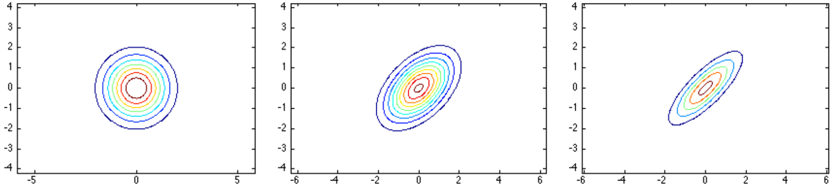


Figure: Contour plot of the pdf

Bivariate Normal

$Cov(x_1, x_2) = 0$ $\quad$ $Cov(x_1, x_2) > 0$ $\quad$ $Cov(x_1, x_2) < 0$

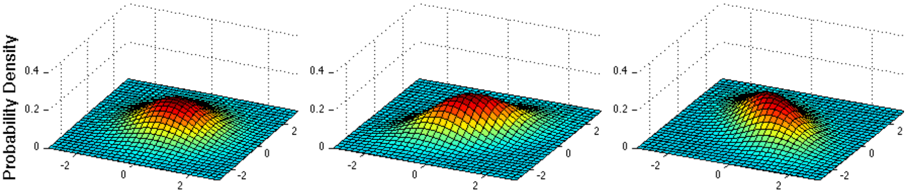


Figure: Probability density function

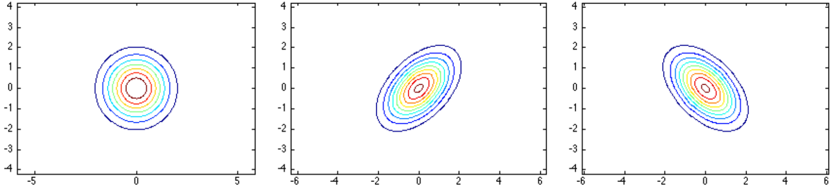


Figure: Contour plot of the pdf

Gaussian Discriminant Analysis (Gaussian Bayes Classifier)

- GDA (GBC) decision boundary is based on class posterior:

$$\begin{aligned}\log p(t_k|\mathbf{x}) &= \log p(\mathbf{x}|t_k) + \log p(t_k) - \log p(\mathbf{x}) \\ &= -\frac{d}{2}\log(2\pi) - \frac{1}{2}\log|\Sigma_k^{-1}| - \frac{1}{2}(\mathbf{x}-\mu_k)^T\Sigma_k^{-1}(\mathbf{x}-\mu_k) + \\ &\quad + \log p(t_k) - \log p(\mathbf{x})\end{aligned}$$

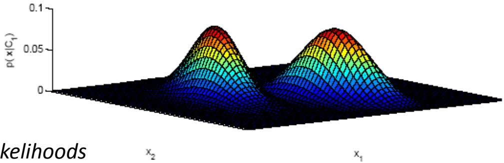
- Decision boundary:

$$(\mathbf{x}-\mu_k)^T\Sigma_k^{-1}(\mathbf{x}-\mu_k) = (\mathbf{x}-\mu_\ell)^T\Sigma_\ell^{-1}(\mathbf{x}-\mu_\ell) + Const$$

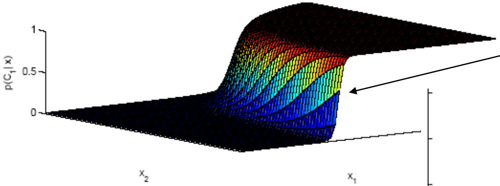
$$\mathbf{x}^T\Sigma_k^{-1}\mathbf{x} - 2\mu_k^T\Sigma_k^{-1}\mathbf{x} = \mathbf{x}^T\Sigma_\ell^{-1}\mathbf{x} - 2\mu_\ell^T\Sigma_\ell^{-1}\mathbf{x} + Const$$

- Quadratic function in $\mathbf{x}$
- What if $\Sigma_k = \Sigma_\ell$?

Decision Boundary

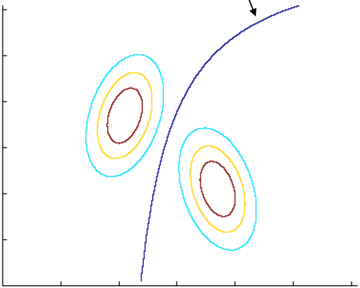


likelihoods



posterior for t_1

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



Learning

- Learn the parameters for each class using maximum likelihood
- Assume the prior is Bernoulli (we have two classes)

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

- You can compute the ML estimate in closed form

$$\begin{aligned}
\phi &= \frac{1}{N}\sum_{n=1}^{N} \mathbb{1}[t^{(n)} = 1] \\
\mu_k &= \frac{\sum_{n=1}^{N} \mathbb{1}[t^{(n)} = k] \cdot \mathbf{x}^{(n)}}{\sum_{n=1}^{N} \mathbb{1}[t^{(n)} = k]} \\
\Sigma_k &= \frac{1}{\sum_{n=1}^{N} \mathbb{1}[t^{(n)} = k]} \sum_{n=1}^{N} \mathbb{1}[t^{(n)} = k](\mathbf{x}^{(n)} - \mu_{t^{(n)}})(\mathbf{x}^{(n)} - \mu_{t^{(n)}})^T
\end{aligned}$$

Simplifying the Model

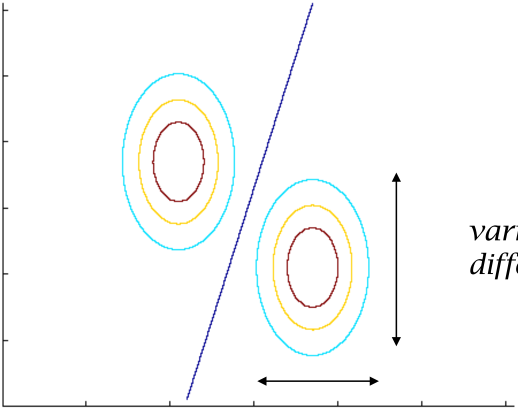
What if $\mathbf{x}$ is high-dimensional?

- For Gaussian Bayes Classifier, if input $\mathbf{x}$ is high-dimensional, then covariance matrix has many parameters
- Save some parameters by using a shared covariance for the classes
- Any other idea you can think of?
- MLE in this case:

$$\Sigma = \frac{1}{N} \sum_{n=1}^{N} (\mathbf{x}^{(n)} - \mu_{t^{(n)}})(\mathbf{x}^{(n)} - \mu_{t^{(n)}})^T$$

- Linear decision boundary.

Decision Boundary: Shared Variances (between Classes)



variances may be different

Gaussian Discriminative Analysis vs Logistic Regression

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

$$p(t|\mathbf{x}, \phi, \mu_0, \mu_1, \Sigma) = \frac{1}{1 + \exp(-\mathbf{w}^T\mathbf{x})}$$

  where $\mathbf{w}$ is an appropriate function of $(\phi, \mu_0, \mu_1, \Sigma)$, $\phi = p(t = 1)$

- Same model as logistic regression!
- When should we prefer GDA to LR, and vice versa?

Gaussian Discriminative Analysis vs Logistic Regression

- GDA makes 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)
- But LR is more robust, less sensitive to incorrect modeling assumptions (what loss is it optimizing?)
- Many class-conditional distributions lead to logistic classifier
- When these distributions are non-Gaussian (a.k.a almost always), LR usually beats GDA
- GDA can handle easily missing features (how do you do that with LR?)

Naive Bayes

- **Naive Bayes**: Assumes features independent given the class

$$p(\mathbf{x}|t = k) = \prod_{i=1}^{d} p(x_i|t = k)$$

- Assuming likelihoods are Gaussian, how many parameters required for Naive Bayes classifier?
- Equivalent to assuming Σ_k is diagonal.

Gaussian Naive Bayes

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

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

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

- Model the same as Gaussian Discriminative Analysis with diagonal covariance matrix

- Maximum likelihood estimate of parameters

$$\begin{aligned} \mu_{ik} &= \frac{\sum_{n=1}^{N} \mathbb{1}[t^{(n)}=k] \cdot x_i^{(n)}}{\sum_{n=1}^{N} \mathbb{1}[t^{(n)}=k]} \\ \sigma_{ik}^2 &= \frac{\sum_{n=1}^{N} \mathbb{1}[t^{(n)}=k] \cdot (x_i^{(n)}-\mu_{ik})^2}{\sum_{n=1}^{N} \mathbb{1}[t^{(n)}=k]} \end{aligned}$$

- What decision boundaries do we get?

Decision Boundary: isotropic

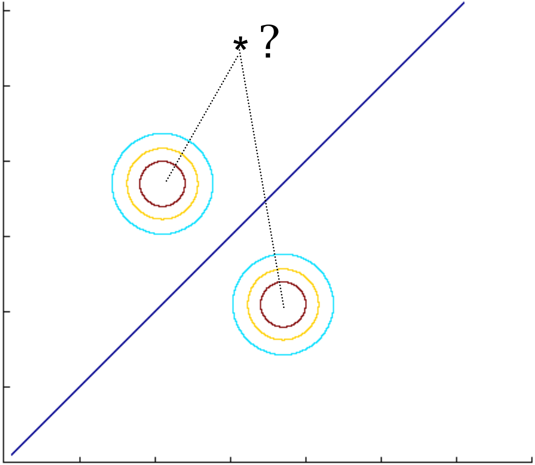
- In this case: $\sigma_{i,k} = \sigma$ (just one parameter), class priors equal (e.g., $p(t_k) = 0.5$ for 2-class case)
- Going back to class posterior for GDA:

$$
\begin{aligned}
\log p(t_k|\mathbf{x}) &= \log p(\mathbf{x}|t_k) + \log p(t_k) - \log p(\mathbf{x}) \\
&= -\frac{d}{2}\log(2\pi) - \frac{1}{2}\log|\Sigma_k^{-1}| - \frac{1}{2}(\mathbf{x}-\mu_k)^T \Sigma_k^{-1}(\mathbf{x}-\mu_k) + \\
&\quad + \log p(t_k) - \log p(\mathbf{x})
\end{aligned}
$$

where we take $\Sigma_k = \sigma^2 I$ and ignore terms that don't depend on k (don't matter when we take max over classes):

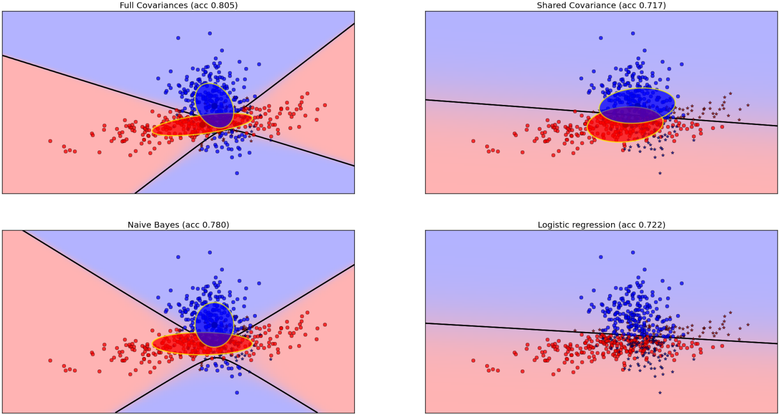
$$
\log p(t_k|\mathbf{x}) = -\frac{1}{2\sigma^2}(\mathbf{x}-\mu_k)^T(\mathbf{x}-\mu_k)
$$

Decision Boundary: isotropic



- Same variance across all classes and input dimensions, all class priors equal
- Classification only depends on distance to the mean. Why?

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 - quadratic decision boundary.
- With shared covariance "collapses" 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 in order to solve a easy problem.