NB-NLP
Naive Bayes for Natural Language Processing

Schwartz

September 21, 2016

How do I love thee? Let me count the Bayes

Types of Bayes

- Empirical Bayes
- **Naive Bayes**
- Full Bayes
- Variational Bayes
- Nonparametric Bayes

Types of priors

- Conjugate prior
- Jeffrey's prior
- Improper prior
- (Un)Informative prior
- Objective prior
- Uniform prior

Types of Markov Chain Monte Carlo (MCMC)

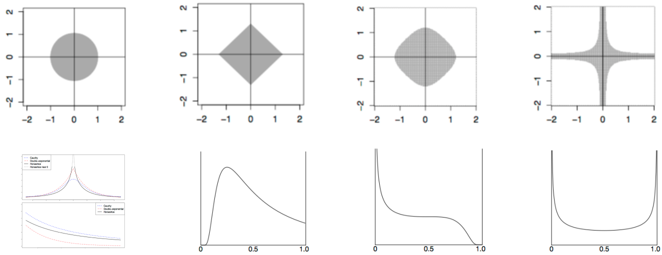
Closed form solutions for posterior distributions are rarely available...

- Gibbs Sampler (cycling through full conditional distributions)
- Metropolis-Hastings (using unnormalized posterior proportionality)
- NUTS: No U-turn sampler (universal probabilistic programming)

Types of Bayesian regularization priors

- Normal-Normal conjugate prior: *ridge regression/regularization*
- Laplace prior: *lasso regularization*
- Cauchy prior: *some other kind of regularization*
- Horseshoe prior: *some other other form of regularization*
  The manuscript presenting the "Horseshoe" prior is entitled "Shrink Globally, Act Locally: Sparse Bayesian Regularization and Prediction"

Gausian | Laplace | Cauchy | Horseshoe



Tails | Implied shrinkage prior profiles from none (0) to total (1) shrinkage

Objectives

1. Understand generative versus predictive modeling

Objectives

1. Understand generative versus predictive modeling
2. Understand "covariance matrix difficulties" when $p > n$

Objectives

1. Understand generative versus predictive modeling
2. Understand "covariance matrix difficulties" when $p > n$
3. Understand how "Naive Bayes" comes to the rescue

Objectives

1. Understand generative versus predictive modeling
2. Understand "covariance matrix difficulties" when $p > n$
3. Understand how "Naive Bayes" comes to the rescue
4. Understand what Naive Bayes classification is & how it works

Objectives

1. Understand generative versus predictive modeling
2. Understand "covariance matrix difficulties" when $p > n$
3. Understand how "Naive Bayes" comes to the rescue
4. Understand what Naive Bayes classification is & how it works
5. Understand how Naive Bayes can be applied to NLP problems

Objectives

1. Understand generative versus predictive modeling
2. Understand "covariance matrix difficulties" when $p > n$
3. Understand how "Naive Bayes" comes to the rescue
4. Understand what Naive Bayes classification is & how it works
5. Understand how Naive Bayes can be applied to NLP problems
6. Know that Naive Bayes is super undemanding computationally

Conditional versus joint models

- Conditional/Predictive/Discriminative
  ("outcome given features")

$$f(Y_i|\boldsymbol{x}_i)$$

Conditional versus joint models

- Conditional/Predictive/Discriminative
  (“outcome given features”)

$$f(Y_i|\boldsymbol{x}_i)$$

- Joint $\rightarrow$ Generative (“features given outcome”)

$$f(Y_i, \boldsymbol{X}_i) \rightarrow f(\boldsymbol{X}_i|Y_i)$$

Conditional versus joint models

- Conditional/Predictive/Discriminative
  ("outcome given features")

$$f(Y_i|\boldsymbol{x}_i)$$

- Joint → Generative ("features given outcome")

$$f(Y_i, \boldsymbol{X}_i) \to f(\boldsymbol{X}_i|Y_i)$$

For categorical $Y_i \in \{k : k = 1, 2, \cdots K\}$

$$\begin{aligned} f(Y_i, \boldsymbol{X}_i) = & \sum_{k=1}^{K} \Pr(Y_i = k) f(\boldsymbol{X}_i|Y_i = k) \\ & \sum_{k=1}^{K} \pi_k f_k(\boldsymbol{X}_i) \\ \implies \quad & f(\boldsymbol{X}_i|Y_i = k) \equiv f_k(\boldsymbol{X}_i) \end{aligned}$$

Conditional versus joint models

- Conditional/Predictive/Discriminative
  ("outcome given features")

$$f(Y_i|\boldsymbol{x}_i)$$

- Joint $\rightarrow$ Generative ("features given outcome")

$$f(Y_i, \boldsymbol{X}_i) \rightarrow f(\boldsymbol{X}_i|Y_i)$$

For categorical $Y_i \in \{k : k = 1, 2, \cdots K\}$

$$\begin{aligned} f(Y_i, \boldsymbol{X}_i) = & \sum_{k=1}^{K} \Pr(Y_i = k) f(\boldsymbol{X}_i|Y_i = k) \\ & \sum_{k=1}^{K} \pi_k f_k(\boldsymbol{X}_i) \\ \implies \quad & f(\boldsymbol{X}_i|Y_i = k) \equiv f_k(\boldsymbol{X}_i) \end{aligned}$$

So we want to model $\boldsymbol{X}_i$'s...

Covariance Matrices

- $\boldsymbol{X}_i \sim MVN(\boldsymbol{\mu}_p, \Sigma_{p\times p})$

  i.e.,

$$\boldsymbol{X}_i \sim MVN\left(\begin{bmatrix} \mu_{X_1} \\ \mu_{X_2} \\ \vdots \\ \mu_{X_p} \end{bmatrix}, \begin{bmatrix} \sigma^2_{X_1} & \sigma_{X_1X_2} & \cdots & \sigma_{X_1X_p} \\ \sigma_{X_2X_1} & \sigma^2_{X_2} & \cdots & \sigma_{X_2X_p} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{X_pX_1} & \sigma_{X_pX_2} & \cdots & \sigma^2_{X_p} \end{bmatrix}\right)$$

Covariance Matrices

- $\boldsymbol{X}_i \sim MVN(\boldsymbol{\mu}_p, \Sigma_{p\times p})$
  i.e.,

$$\boldsymbol{X}_i \sim MVN\left(\begin{bmatrix} \mu_{X_1} \\ \mu_{X_2} \\ \vdots \\ \mu_{X_p} \end{bmatrix}, \begin{bmatrix} \sigma^2_{X_1} & \sigma_{X_1X_2} & \cdots & \sigma_{X_1X_p} \\ \sigma_{X_2X_1} & \sigma^2_{X_2} & \cdots & \sigma_{X_2X_p} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{X_pX_1} & \sigma_{X_pX_2} & \cdots & \sigma^2_{X_p} \end{bmatrix}\right)$$

- $\hat{\Sigma}_{p\times p} = \frac{\sum_{i=1}^{n}(\boldsymbol{X}_i - \bar{\boldsymbol{X}})_{p\times 1}(\boldsymbol{X}_i - \bar{\boldsymbol{X}})^T_{p\times 1}}{n-1}$

Covariance Matrices

- $\boldsymbol{X}_i \sim MVN(\boldsymbol{\mu}_p, \Sigma_{p\times p})$
  i.e.,

$$\boldsymbol{X}_i \sim MVN\left(\begin{bmatrix} \mu_{X_1} \\ \mu_{X_2} \\ \vdots \\ \mu_{X_p} \end{bmatrix}, \begin{bmatrix} \sigma^2_{X_1} & \sigma_{X_1X_2} & \cdots & \sigma_{X_1X_p} \\ \sigma_{X_2X_1} & \sigma^2_{X_2} & \cdots & \sigma_{X_2X_p} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{X_pX_1} & \sigma_{X_pX_2} & \cdots & \sigma^2_{X_p} \end{bmatrix}\right)$$

- $\hat{\Sigma}_{p\times p} = \frac{\sum_{i=1}^{n}(\boldsymbol{X}_i - \bar{\boldsymbol{X}})_{p\times 1}(\boldsymbol{X}_i - \bar{\boldsymbol{X}})^T_{p\times 1}}{n-1}$
- If $n < p$, the *rank* of $\hat{\Sigma}_{p\times p}$ is n
  since it is a linear combination of only with n *independent* $\boldsymbol{X}_i's$

Covariance Matrices

- $\boldsymbol{X}_i \sim MVN(\boldsymbol{\mu}_p, \Sigma_{p\times p})$
  i.e.,

$$\boldsymbol{X}_i \sim MVN\left(\begin{bmatrix} \mu_{X_1} \\ \mu_{X_2} \\ \vdots \\ \mu_{X_p} \end{bmatrix}, \begin{bmatrix} \sigma^2_{X_1} & \sigma_{X_1X_2} & \cdots & \sigma_{X_1X_p} \\ \sigma_{X_2X_1} & \sigma^2_{X_2} & \cdots & \sigma_{X_2X_p} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{X_pX_1} & \sigma_{X_pX_2} & \cdots & \sigma^2_{X_p} \end{bmatrix}\right)$$

- $\hat{\Sigma}_{p\times p} = \frac{\sum_{i=1}^{n}(\boldsymbol{X}_i - \bar{\boldsymbol{X}})_{p\times 1}(\boldsymbol{X}_i - \bar{\boldsymbol{X}})^T_{p\times 1}}{n-1}$
- If $n < p$, the *rank* of $\hat{\Sigma}_{p\times p}$ is n
  since it is a linear combination of only with *n independent* $\boldsymbol{X}_i'$*s*
- Thus, $\hat{\Sigma}_{p\times p}$ *will not* be invertible

Covariance Matrices

- $\boldsymbol{X}_i \sim MVN(\boldsymbol{\mu}_p, \Sigma_{p\times p})$
  i.e.,

$$\boldsymbol{X}_i \sim MVN\left(\begin{bmatrix} \mu_{X_1} \\ \mu_{X_2} \\ \vdots \\ \mu_{X_p} \end{bmatrix}, \begin{bmatrix} \sigma^2_{X_1} & \sigma_{X_1X_2} & \cdots & \sigma_{X_1X_p} \\ \sigma_{X_2X_1} & \sigma^2_{X_2} & \cdots & \sigma_{X_2X_p} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{X_pX_1} & \sigma_{X_pX_2} & \cdots & \sigma^2_{X_p} \end{bmatrix}\right)$$

- $\hat{\Sigma}_{p\times p} = \frac{\sum_{i=1}^{n}(\boldsymbol{X}_i - \bar{\boldsymbol{X}})_{p\times 1}(\boldsymbol{X}_i - \bar{\boldsymbol{X}})^T_{p\times 1}}{n-1}$
- If $n < p$, the *rank* of $\hat{\Sigma}_{p\times p}$ is n
  since it is a linear combination of only with *n independent* $\boldsymbol{X}_i'$*s*
- Thus, $\hat{\Sigma}_{p\times p}$ *will not* be invertible
- This is bad news since the pdf of $\boldsymbol{X} \sim MVN(\boldsymbol{\mu}, \Sigma)$ is

$$(2\pi)^{-\frac{k}{2}}|\Sigma|^{-1}e^{-\frac{1}{2}}(\boldsymbol{X} - \boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{X} - \boldsymbol{\mu})$$

Covariance Matrices

- $\boldsymbol{X}_i \sim MVN(\boldsymbol{\mu}_p, \Sigma_{p\times p})$
  i.e.,

$$\boldsymbol{X}_i \sim MVN\left(\begin{bmatrix} \mu_{X_1} \\ \mu_{X_2} \\ \vdots \\ \mu_{X_p} \end{bmatrix}, \begin{bmatrix} \sigma^2_{X_1} & 0 & \cdots & 0 \\ 0 & \sigma^2_{X_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma^2_{X_p} \end{bmatrix} \right)$$

- $\hat{\Sigma}_{p\times p} = \frac{\sum_{i=1}^{n}(\boldsymbol{X}_i - \bar{\boldsymbol{X}})_{p\times 1}(\boldsymbol{X}_i - \bar{\boldsymbol{X}})^T_{p\times 1}}{n-1}$
- If $n < p$, the *rank* of $\hat{\Sigma}_{p\times p}$ is n
  since it is a linear combination of only with *n independent $\boldsymbol{X}_i'$s*
- Thus, $\hat{\Sigma}_{p\times p}$ *will not* be invertible
- This is bad news since the pdf of $\boldsymbol{X} \sim MVN(\boldsymbol{\mu}, \Sigma)$ is

$$(2\pi)^{-\frac{k}{2}}|\Sigma|^{-1}e^{-\frac{1}{2}}(\boldsymbol{X} - \boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{X} - \boldsymbol{\mu})$$

Covariance Matrices

- $\boldsymbol{X}_i \sim MVN(\boldsymbol{\mu}_p, \Sigma_{p\times p})$
  i.e.,

$$\boldsymbol{X}_i \sim MVN\left(\begin{bmatrix} \mu_{X_1} \\ \mu_{X_2} \\ \vdots \\ \mu_{X_p} \end{bmatrix}, \begin{bmatrix} \sigma^2_{X_1} & 0 & \cdots & 0 \\ 0 & \sigma^2_{X_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma^2_{X_p} \end{bmatrix}\right)$$

- $\sigma^2_{X_j} = \frac{\sum_{i=1}^{n}(X_{ji}-\mu_j)^2}{n-1}$ and $\sigma_{X_j X_k} = 0$ for $j \neq k$
- If $n < p$, the *rank* of $\hat{\Sigma}_{p\times p}$ is n
  since it is a linear combination of only with *n independent $\boldsymbol{X}_i'$s*
- Thus, $\hat{\Sigma}_{p\times p}$ *will not* be invertible
- This is bad news since the pdf of $\boldsymbol{X} \sim MVN(\boldsymbol{\mu}, \Sigma)$ is

$$(2\pi)^{-\frac{k}{2}}|\Sigma|^{-1}e^{-\frac{1}{2}}(\boldsymbol{X}-\boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{X}-\boldsymbol{\mu})$$

Covariance Matrices

- $\boldsymbol{X}_i \sim MVN(\boldsymbol{\mu}_p, \Sigma_{p\times p})$
  i.e.,

$$\boldsymbol{X}_i \sim MVN\left(\begin{bmatrix} \mu_{X_1} \\ \mu_{X_2} \\ \vdots \\ \mu_{X_p} \end{bmatrix}, \begin{bmatrix} \sigma^2_{X_1} & 0 & \cdots & 0 \\ 0 & \sigma^2_{X_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma^2_{X_p} \end{bmatrix} \right)$$

- $\sigma^2_{X_j} = \frac{\sum_{i=1}^n (X_{ji}-\mu_j)^2}{n-1}$ and $\sigma_{X_j X_k} = 0$ for $j \neq k$
- If $n < p$, the *rank* of $\hat{\Sigma}_{p\times p}$ is $\equiv p$
  because it's a $p \times p$ invertible diagonal matrix
- Thus, $\hat{\Sigma}_{p\times p}$ *will not* be invertible
- This is bad news since the pdf of $\boldsymbol{X} \sim MVN(\boldsymbol{\mu}, \Sigma)$ is

$$(2\pi)^{-\frac{k}{2}} |\Sigma|^{-1} e^{-\frac{1}{2}} (\boldsymbol{X} - \boldsymbol{\mu})^T \Sigma^{-1} (\boldsymbol{X} - \boldsymbol{\mu})$$

Covariance Matrices

- $\boldsymbol{X}_i \sim MVN(\boldsymbol{\mu}_p, \Sigma_{p\times p})$
  i.e.,

$$\boldsymbol{X}_i \sim MVN\left(\begin{bmatrix} \mu_{X_1} \\ \mu_{X_2} \\ \vdots \\ \mu_{X_p} \end{bmatrix}, \begin{bmatrix} \sigma^2_{X_1} & 0 & \cdots & 0 \\ 0 & \sigma^2_{X_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma^2_{X_p} \end{bmatrix}\right)$$

- $\sigma^2_{X_j} = \frac{\sum_{i=1}^{n}(X_{ji}-\mu_j)^2}{n-1}$ and $\sigma_{X_jX_k} = 0$ for $j \neq k$
- If $n < p$, the *rank* of $\hat{\Sigma}_{p\times p}$ is $\neq p$
  because it's a $p \times p$ invertible diagonal matrix
- Thus, $\hat{\Sigma}_{p\times p}$ *will* **be invertible**
- This is bad news since the pdf of $\boldsymbol{X} \sim MVN(\boldsymbol{\mu}, \Sigma)$ is

$$(2\pi)^{-\frac{k}{2}}|\Sigma|^{-1}e^{-\frac{1}{2}}(\boldsymbol{X}-\boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{X}-\boldsymbol{\mu})$$

Covariance Matrices

- $\boldsymbol{X}_i \sim MVN(\boldsymbol{\mu}_p, \Sigma_{p \times p})$
  i.e.,

$$\boldsymbol{X}_i \sim MVN\left(\begin{bmatrix} \mu_{X_1} \\ \mu_{X_2} \\ \vdots \\ \mu_{X_p} \end{bmatrix}, \begin{bmatrix} \sigma^2_{X_1} & 0 & \cdots & 0 \\ 0 & \sigma^2_{X_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma^2_{X_p} \end{bmatrix} \right)$$

- $\sigma^2_{X_j} = \frac{\sum_{i=1}^n (X_{ji} - \mu_j)^2}{n-1}$ and $\sigma_{X_j X_k} = 0$ for $j \neq k$
- If $n < p$, the *rank* of $\hat{\Sigma}_{p \times p}$ is $= p$
  because it's a $p \times p$ invertible diagonal matrix
- Thus, $\hat{\Sigma}_{p \times p}$ *will* **be invertible**
- This is good news since the pdf of $\boldsymbol{X} \sim MVN(\boldsymbol{\mu}, \Sigma)$ is

$$(2\pi)^{-\frac{k}{2}} |\Sigma|^{-1} e^{-\frac{1}{2}} (\boldsymbol{X} - \boldsymbol{\mu})^T \Sigma^{-1} (\boldsymbol{X} - \boldsymbol{\mu})$$

Time for some explicit clarification of the notation

Because there's a lot going on here and it *is confusing*...

Time for some explicit clarification of the notation

Because there's a lot going on here and it *is confusing*...

You need to be *very careful* to keep everything straight

Time for some explicit clarification of the notation

Because there's a lot going on here and it *is confusing*...

You need to be *very careful* to keep everything straight

- $\boldsymbol{X}_i$ is a vector p of features related to outcome Y_i

Time for some explicit clarification of the notation

Because there's a lot going on here and it *is confusing*...

You need to be *very careful* to keep everything straight

- $\boldsymbol{X}_i$ is a vector p of features related to outcome Y_i
- Each of the p features in $\boldsymbol{X}_i$ is referred to as X_j

Time for some explicit clarification of the notation

Because there's a lot going on here and it *is confusing*...

You need to be *very careful* to keep everything straight

- $\boldsymbol{X}_i$ is a vector p of features related to outcome Y_i
- Each of the p features in $\boldsymbol{X}_i$ is referred to as X_j
- A specific value for feature X_j of outcome Y_i is notated as X_{ji}

Time for some explicit clarification of the notation

Because there's a lot going on here and it *is confusing*...

You need to be *very careful* to keep everything straight

- $\boldsymbol{X}_i$ is a vector p of features related to outcome Y_i
- Each of the p features in $\boldsymbol{X}_i$ is referred to as X_j
- A specific value for feature X_j of outcome Y_i is notated as X_{ji}
- Outcome $Y_i \in \{k : k = 1, 2, \cdots K\}$ is categorial, taking on one of K possible outcome values referred to as k

Estimating the joint distribution of features and outcomes

$$f(Y_i, \boldsymbol{X}_i) = \sum_{k=1}^{K} \pi_k f_k(\boldsymbol{X}_i)$$

Estimating the joint distribution of features and outcomes

$$f(Y_i, \boldsymbol{X}_i) = \sum_{k=1}^{K} \pi_k f_k(\boldsymbol{X}_i)$$

- Estimate $\pi_k \equiv \Pr(Y_i = k)$ with $\frac{1}{n}\sum 1_{[Y_i = k]}$

Estimating the joint distribution of features and outcomes

$$f(Y_i, \boldsymbol{X}_i) = \sum_{k=1}^{K} \pi_k f_k(\boldsymbol{X}_i)$$

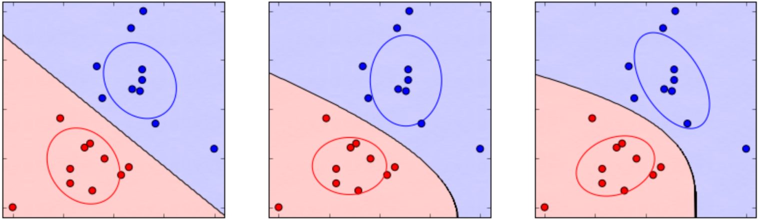
- Estimate $\pi_k \equiv \Pr(Y_i = k)$ with $\frac{1}{n} \sum 1_{[Y_i = k]}$
- Estimate $f_k(\boldsymbol{X}_i) \equiv f(\boldsymbol{X}_i | Y_i = k)$ with

$$\boldsymbol{X}_i \sim MVN \left(\begin{bmatrix} \hat{\mu}_{k X_1} \\ \hat{\mu}_{k X_2} \\ \vdots \\ \mu_{k X_p} \end{bmatrix}, \begin{bmatrix} \hat{\sigma}^2_{k X_1} & 0 & \cdots & 0 \\ 0 & \hat{\sigma}^2_{k X_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \hat{\sigma}^2_{k X_p} \end{bmatrix} \right)$$

where $\hat{\sigma}^2_{k X_j} = \frac{\sum_{i=1: Y_i = k}^{n} (X_{ji} - \bar{X}_j)^2}{\left(\sum_{i=1: Y_i = k}^{n} 1 \right) - 1}$ and $\hat{\mu}_{k X_j} = \frac{\sum_{i=1: Y_i = k}^{n} X_{ji}}{\sum_{i=1: Y_i = k}^{n} 1}$

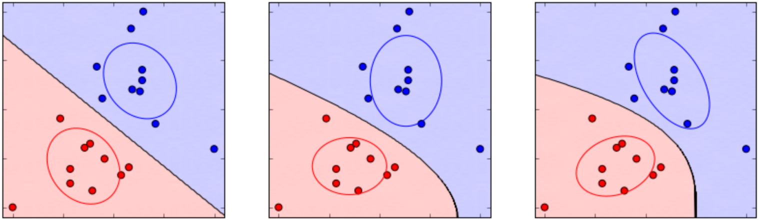
What is the assumption on the covariance matrix doing?

$$\begin{bmatrix} \hat{\sigma}^2_{kX_1} & 0 & \cdots & 0 \\ 0 & \hat{\sigma}^2_{kX_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \hat{\sigma}^2_{kX_p} \end{bmatrix}$$



What is the assumption on the covariance matrix doing?

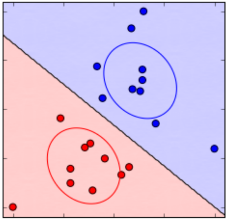
$$\begin{bmatrix} \hat{\sigma}^2_{kX_1} & 0 & \cdots & 0 \\ 0 & \hat{\sigma}^2_{kX_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \hat{\sigma}^2_{kX_p} \end{bmatrix}$$

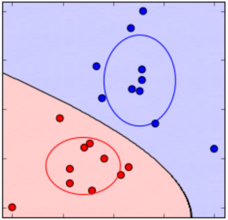


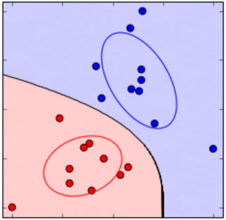
Linear Discriminant Analysis (LDA) — Naive Bayes — Quadratic Discriminant Analysis (QDA)

What is the assumption on the covariance matrix doing?

$$\begin{bmatrix} \hat{\sigma}^2_{kX_1} & 0 & \cdots & 0 \\ 0 & \hat{\sigma}^2_{kX_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \hat{\sigma}^2_{kX_p} \end{bmatrix}$$


Linear Discriminant Analysis (LDA)


Naive Bayes


Quadratic Discriminant Analysis (QDA)

$\Uparrow\Uparrow\Uparrow\Uparrow\Uparrow$

Why is this even called Bayes?

$$\Pr(Y_i = k | \boldsymbol{X}_i)$$

Why is this even called Bayes?

$$\begin{aligned}
&\Pr(Y_i = k|\boldsymbol{X}_i) \\
= \; &\frac{f(\boldsymbol{X}_i|Y_i = k)\Pr(Y = k)}{f(\boldsymbol{X}_i)}
\end{aligned}$$

Why is this even called Bayes?

$$
\begin{aligned}
&\Pr(Y_i = k|\boldsymbol{X}_i) \\
&= \frac{f(\boldsymbol{X}_i|Y_i = k)\Pr(Y = k)}{f(\boldsymbol{X}_i)}
\end{aligned}
$$

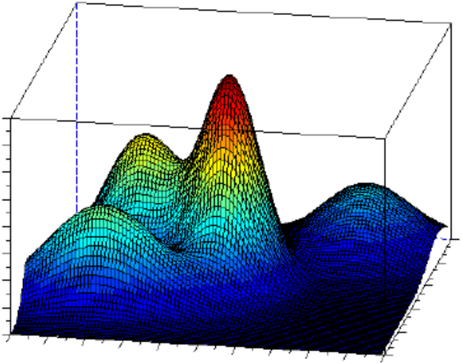
$$
= \frac{\pi_k f_k(\boldsymbol{X}_i)}{\sum_{k'=1}^{K} \pi_{k'} f_{k'}(\boldsymbol{X}_i)}
$$

Why is this even called Bayes?

$$
\begin{aligned}
& \Pr(Y_i = k | \boldsymbol{X}_i) \\
= & \frac{f(\boldsymbol{X}_i | Y_i = k) \Pr(Y = k)}{f(\boldsymbol{X}_i)} \\
= & \frac{\pi_k f_k(\boldsymbol{X}_i)}{\sum_{k'=1}^{K} \pi_{k'} f_{k'}(\boldsymbol{X}_i)} \\
\propto & \, \pi_k f_k(\boldsymbol{X}_i)
\end{aligned}
$$

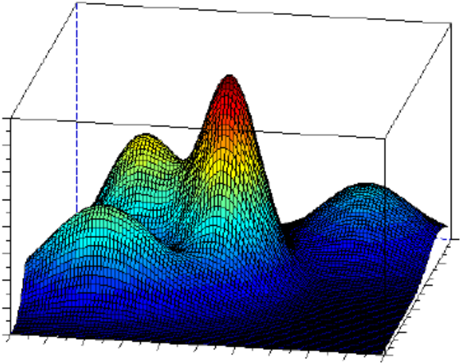
Why is this even called Bayes?

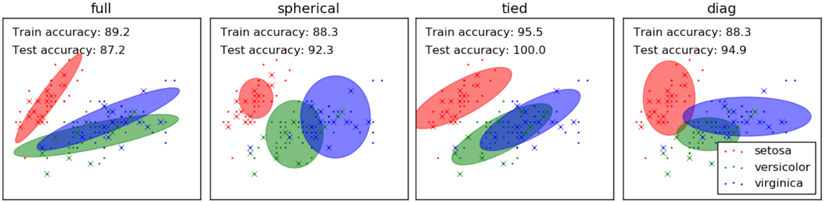
$$
\begin{aligned}
&\Pr(Y_i = k|\boldsymbol{X}_i) \\
&= \frac{f(\boldsymbol{X}_i|Y_i = k)\Pr(Y = k)}{f(\boldsymbol{X}_i)} \\
&= \frac{\pi_k f_k(\boldsymbol{X}_i)}{\sum_{k'=1}^{K} \pi_{k'} f_{k'}(\boldsymbol{X}_i)} \\
&\propto \pi_k f_k(\boldsymbol{X}_i)
\end{aligned}
$$



Why is this even called Bayes?

$$
\begin{aligned}
&\Pr(Y_i = k|\boldsymbol{X}_i) \\
&= \frac{f(\boldsymbol{X}_i|Y_i = k)\Pr(Y = k)}{f(\boldsymbol{X}_i)} \\
&= \frac{\pi_k f_k(\boldsymbol{X}_i)}{\sum_{k'=1}^{K} \pi_{k'} f_{k'}(\boldsymbol{X}_i)} \\
&\propto \pi_k f_k(\boldsymbol{X}_i)
\end{aligned}
$$





Other models for feature X_j

- So far we've assumed that feature X_j is

Other models for feature X_j

- So far we've assumed that feature X_j is
  (a) continuous valued and *normally distributed*

Other models for feature X_j

- So far we've assumed that feature X_j is
  (a) continuous valued and *normally distributed*
  (b) independent of the other features $X_{j'}$ for $j' \neq j$

Other models for feature X_j

- So far we've assumed that feature X_j is
  (a) continuous valued and *normally distributed*
  (b) independent of the other features $X_{j'}$ for $j' \neq j$

  I.e., we said $\boldsymbol{X}_i$ is *MVN* with a diagonal covariance matrix

Other models for feature X_j

- So far we've assumed that feature X_j is
  (a) continuous valued and *normally distributed*
  (b) independent of the other features $X_{j'}$ for $j' \neq j$

  I.e., we said $\boldsymbol{X}_i$ is *MVN* with a diagonal covariance matrix
- But we can also have features X_j that are

Other models for feature X_j

- So far we've assumed that feature X_j is
  (a) continuous valued and *normally distributed*
  (b) independent of the other features $X_{j'}$ for $j' \neq j$

  I.e., we said $\boldsymbol{X}_i$ is *MVN* with a diagonal covariance matrix
- But we can also have features X_j that are

  Multinomial for categorical X_j (e.g., counts of words in doc)

Other models for feature X_j

- So far we've assumed that feature X_j is
  (a) continuous valued and *normally distributed*
  (b) independent of the other features $X_{j'}$ for $j' \neq j$

  I.e., we said $\boldsymbol{X}_i$ is *MVN* with a diagonal covariance matrix
- But we can also have features X_j that are

  Multinomial for categorical X_j (e.g., counts of words in doc)

  Bernoulli for indicator X_j (e.g., appearance of word in doc)

  both of these assume features (e.g., words) are independent

Other models for feature X_j

- So far we've assumed that feature X_j is
  (a) continuous valued and *normally distributed*
  (b) independent of the other features $X_{j'}$ for $j' \neq j$

  I.e., we said $\boldsymbol{X}_i$ is *MVN* with a diagonal covariance matrix
- But we can also have features X_j that are

  Multinomial for categorical X_j (e.g., counts of words in doc)

  Bernoulli for indicator X_j (e.g., appearance of word in doc)

  both of these assume features (e.g., words) are independent

  And so on... depending on our choices for modeling X_j

Other models for feature X_j

- So far we've assumed that feature X_j is
  (a) continuous valued and *normally distributed*
  (b) independent of the other features $X_{j'}$ for $j' \neq j$

  I.e., we said $\boldsymbol{X}_i$ is *MVN* with a diagonal covariance matrix
- But we can also have features X_j that are

  Multinomial for categorical X_j (e.g., counts of words in doc)

  Bernoulli for indicator X_j (e.g., appearance of word in doc)

  both of these assume features (e.g., words) are independent

  And so on... depending on our choices for modeling X_j
- Can we use different types of X_j at the same time? Of course

Other models for feature X_j

- So far we've assumed that feature X_j is
  (a) continuous valued and *normally distributed*
  (b) independent of the other features $X_{j'}$ for $j' \neq j$

  I.e., we said $\boldsymbol{X}_i$ is *MVN* with a diagonal covariance matrix
- But we can also have features X_j that are

  Multinomial for categorical X_j (e.g., counts of words in doc)

  Bernoulli for indicator X_j (e.g., appearance of word in doc)

  both of these assume features (e.g., words) are independent

  And so on... depending on our choices for modeling X_j
- Can we use different types of X_j at the same time? Of course

$$f_k(\boldsymbol{X}_i) = \prod_{j=1}^{p} f_k(X_{ji})$$

Other models for feature X_j

- So far we've assumed that feature X_j is
  (a) continuous valued and *normally distributed*
  (b) independent of the other features $X_{j'}$ for $j' \neq j$

  I.e., we said $\boldsymbol{X}_i$ is *MVN* with a diagonal covariance matrix
- But we can also have features X_j that are

  Multinomial for categorical X_j (e.g., counts of words in doc)

  Bernoulli for indicator X_j (e.g., appearance of word in doc)

  both of these assume features (e.g., words) are independent

  And so on... depending on our choices for modeling X_j
- Can we use different types of X_j at the same time? Of course

$$f_k(\boldsymbol{X}_i) = \prod_{j=1}^{p} f_k(X_{ji})$$

As long as we assume the features are independent...

Other models for feature X_j

- So far we've assumed that feature X_j is
  - (a) continuous valued and *normally distributed*
  - (b) independent of the other features $X_{j'}$ for $j' \neq j$

  I.e., we said $\boldsymbol{X}_i$ is *MVN* with a diagonal covariance matrix
- But we can also have features X_j that are

  Multinomial for categorical X_j (e.g., counts of words in doc)

  Bernoulli for indicator X_j (e.g., appearance of word in doc)

  both of these assume features (e.g., words) are independent

  And so on... depending on our choices for modeling X_j
- Can we use different types of X_j at the same time? Of course

$$f_k(\boldsymbol{X}_i) = \prod_{j=1}^{p} f_k(X_{ji})$$

As long as we assume the features are independent...

That's actually exactly what we did with the diagonal *MVN* :

Other models for feature X_j

- So far we've assumed that feature X_j is
  (a) continuous valued and *normally distributed*
  (b) independent of the other features $X_{j'}$ for $j' \neq j$

  I.e., we said $\boldsymbol{X}_i$ is MVN with a diagonal covariance matrix
- But we can also have features X_j that are

  Multinomial for categorical X_j (e.g., counts of words in doc)

  Bernoulli for indicator X_j (e.g., appearance of word in doc)

  both of these assume features (e.g., words) are independent

  And so on... depending on our choices for modeling X_j
- Can we use different types of X_j at the same time? Of course

$$f_k(\boldsymbol{X}_i) = \prod_{j=1}^{p} f_k(X_{ji})$$

  As long as we assume the features are independent...

  That's actually exactly what we did with the diagonal MVN : assumed X_j's independent normals & multiplied $\Longrightarrow MVN$

And don't forget...

$$\Pr(Y_i = k | \boldsymbol{X}_i) \propto \pi_k f_k(\boldsymbol{X}_i) = \pi_k \prod_{j=1}^{p} f_k(X_{ji})$$

multiply, multiply, multiply – it's that easy

And don't forget...

$$\Pr(Y_i = k | \boldsymbol{X}_i) \propto \pi_k f_k(\boldsymbol{X}_i) = \pi_k \prod_{j=1}^{p} f_k(X_{ji})$$

multiply, multiply, multiply – it's that easy

- E.g., in classifying news story types

$$\hat{\pi}_k = \frac{\text{number of sports articles}}{\text{total number of articles}}$$

And don't forget...

$$\Pr(Y_i = k|\boldsymbol{X}_i) \propto \pi_k f_k(\boldsymbol{X}_i) = \pi_k \prod_{j=1}^{p} f_k(X_{ji})$$

multiply, multiply, multiply – it's that easy

- E.g., in classifying news story types

$$\hat{\pi}_k = \frac{\text{number of sports articles}}{\text{total number of articles}}$$

f_{sports}("Cleveland's 52-year championship drought ended with the 2015/16 NBA season...")

And don't forget...

$$\Pr(Y_i = k | \boldsymbol{X}_i) \propto \pi_k f_k(\boldsymbol{X}_i) = \pi_k \prod_{j=1}^{p} f_k(X_{ji})$$

multiply, multiply, multiply – it's that easy

- E.g., in classifying news story types

$$\hat{\pi}_k = \frac{\text{number of sports articles}}{\text{total number of articles}}$$

$$f_{\text{sports}}(\text{“Cleveland's 52-year championship drought ended with the 2015/16 NBA season...”})$$

$$= \Pr(\text{“Cleveland's”}|\text{sports}) \Pr(\text{“52-year”}|\text{sports}) \Pr(\text{“Championship”}|\text{sports}) \cdots$$

And don't forget...

$$\Pr(Y_i = k|\boldsymbol{X}_i) \propto \pi_k f_k(\boldsymbol{X}_i) = \pi_k \prod_{j=1}^{p} f_k(X_{ji})$$

multiply, multiply, multiply – it's that easy

- E.g., in classifying news story types

$$\hat{\pi}_k = \frac{\text{number of sports articles}}{\text{total number of articles}}$$

$$f_{\text{sports}}(\text{“Cleveland's 52-year championship drought ended with the 2015/16 NBA season...”})$$

$$= \Pr(\text{“Cleveland's”}|\text{sports}) \Pr(\text{“52-year”}|\text{sports}) \Pr(\text{“Championship”}|\text{sports}) \cdots$$

$$= \frac{\#\ \text{“Cleveland's” in sports} + \alpha}{\#\ \text{words in sports} + \alpha \times\ \text{total}\ \#\ \text{words}} \cdot \frac{\#\ \text{“52-year” in sports} + \alpha}{\#\ \text{words in sports} + \alpha \times\ \text{total}\ \#\ \text{words}} \cdots$$

And don't forget...

$$\Pr(Y_i = k|\boldsymbol{X}_i) \propto \pi_k f_k(\boldsymbol{X}_i) = \pi_k \prod_{j=1}^{p} f_k(X_{ji})$$

multiply, multiply, multiply – it's that easy

- E.g., in classifying news story types

$$\hat{\pi}_k = \frac{\text{number of sports articles}}{\text{total number of articles}}$$

$$f_{\text{sports}}(\text{“Cleveland's 52-year championship drought ended with the 2015/16 NBA season...”})$$

$$= \Pr(\text{“Cleveland's”}|\text{sports}) \Pr(\text{“52-year”}|\text{sports}) \Pr(\text{“Championship”}|\text{sports}) \cdots$$

$$= \frac{\#\ \text{“Cleveland's” in sports} + \alpha}{\#\ \text{words in sports} + \alpha \times\ \text{total}\ \#\ \text{words}} \cdot \frac{\#\ \text{“52-year” in sports} + \alpha}{\#\ \text{words in sports} + \alpha \times\ \text{total}\ \#\ \text{words}} \cdots$$

- The α is *Laplace smoothing* – it avoids multiplying by 0

And don't forget...

$$\Pr(Y_i = k | \boldsymbol{X}_i) \propto \pi_k f_k(\boldsymbol{X}_i) = \pi_k \prod_{j=1}^{p} f_k(X_{ji})$$

multiply, multiply, multiply – it's that easy

- E.g., in classifying news story types

$$\hat{\pi}_k = \frac{\text{number of sports articles}}{\text{total number of articles}}$$

$$f_{\text{sports}}(\text{“Cleveland's 52-year championship drought ended with the 2015/16 NBA season...”})$$

$$= \Pr(\text{“Cleveland's”}|\text{sports}) \Pr(\text{“52-year”}|\text{sports}) \Pr(\text{“Championship”}|\text{sports}) \cdots$$

$$= \frac{\#\ \text{“Cleveland's” in sports} + \alpha}{\#\ \text{words in sports} + \alpha \times\ \text{total}\ \#\ \text{words}} \cdot \frac{\#\ \text{“52-year” in sports} + \alpha}{\#\ \text{words in sports} + \alpha \times\ \text{total}\ \#\ \text{words}} \cdots$$

- The α is *Laplace smoothing* – it avoids multiplying by 0
- You'll probably also want to take a *log*...

Parting comments

- Truly correlated features can hamper Naive Bayes (NB)

Parting comments

- Truly correlated features can hamper Naive Bayes (NB)
- Probability estimates are unreliable with the naive assumption

Parting comments

- Truly correlated features can hamper Naive Bayes (NB)
- Probability estimates are unreliable with the naive assumption
- But NB classifications can be workable... but

Parting comments

- Truly correlated features can hamper Naive Bayes (NB)
- Probability estimates are unreliable with the naive assumption
- But NB classifications can be workable... but
- NB is typically outperformed by less naive methodologies

Parting comments

- Truly correlated features can hamper Naive Bayes (NB)
- Probability estimates are unreliable with the naive assumption
- But NB classifications can be workable... but
- NB is typically outperformed by less naive methodologies

However...

Parting comments

- Truly correlated features can hamper Naive Bayes (NB)
- Probability estimates are unreliable with the naive assumption
- But NB classifications can be workable... but
- NB is typically outperformed by less naive methodologies

  However...
- NB is super undemanding computationally:

Parting comments

- Truly correlated features can hamper Naive Bayes (NB)
- Probability estimates are unreliable with the naive assumption
- But NB classifications can be workable... but
- NB is typically outperformed by less naive methodologies

  However...
- NB is super undemanding computationally:

  NB can handle huge data sets very quickly – i.e., in real time

Parting comments

- Truly correlated features can hamper Naive Bayes (NB)
- Probability estimates are unreliable with the naive assumption
- But NB classifications can be workable... but
- NB is typically outperformed by less naive methodologies

  However...
- NB is super undemanding computationally:

  NB can handle huge data sets very quickly – i.e., in real time

  NB can handle wide data sets other methodologies can't...

Parting comments

- Truly correlated features can hamper Naive Bayes (NB)
- Probability estimates are unreliable with the naive assumption
- But NB classifications can be workable... but
- NB is typically outperformed by less naive methodologies

  However...
- NB is super undemanding computationally:

  NB can handle huge data sets very quickly – i.e., in real time

  NB can handle wide data sets other methodologies can't...
- And NB is very simple to implement and use...

Parting comments

- Truly correlated features can hamper Naive Bayes (NB)
- Probability estimates are unreliable with the naive assumption
- But NB classifications can be workable... but
- NB is typically outperformed by less naive methodologies

  However...
- NB is super undemanding computationally:

  NB can handle huge data sets very quickly – i.e., in real time

  NB can handle wide data sets other methodologies can't...
- And NB is very simple to implement and use...

  Although isn't *everything* in scikit-learn?