Decision boundary $\{x : P(Y=1|X=x) = P(Y=0|X=x)\}$

d-dim Gaussian Bayes classifier

$X = \begin{bmatrix} X_1 \\ \vdots \\ X_d \end{bmatrix}$

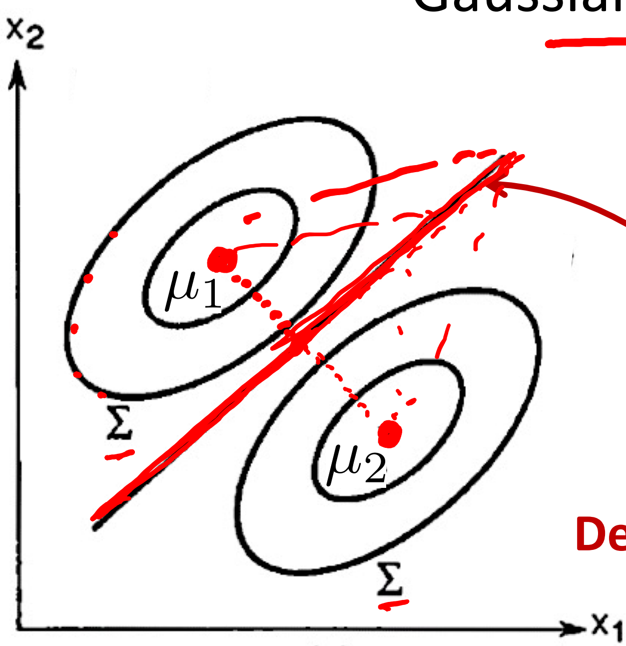
$$f(X) = \arg\max_{Y=y} \underbrace{P(X=x|Y=y)}_{\text{Class conditional Distribution of features}} \underbrace{P(Y=y)}_{\text{Class distribution}}$$

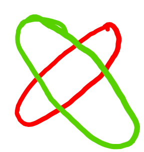
- What decision boundaries can we get in d-dim?

Gaussian(μ_y,Σ_y)

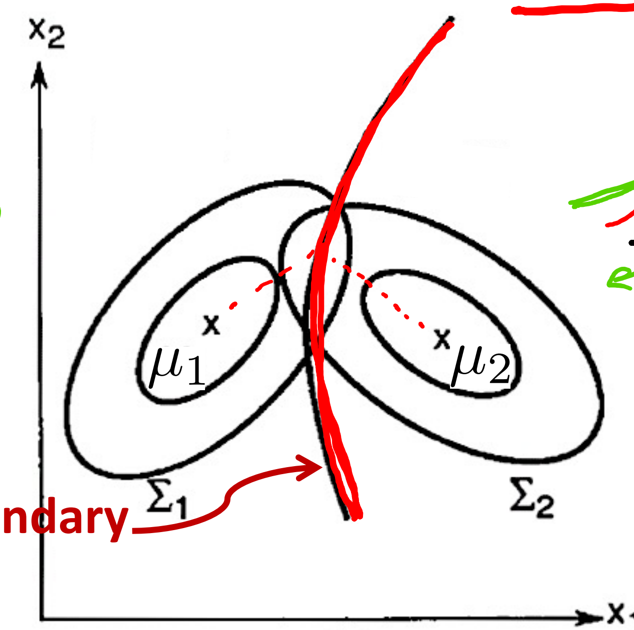
$\frac{1}{2}$

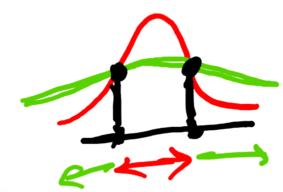
Bernoulli(θ)

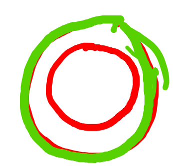




Decision Boundary







Decision Boundary of Gaussian Bayes

- Decision boundary is set of points x: P(Y=1|X=x) = P(Y=0|X=x)
- By Bayes theorem, equivalent to x:

$$\{x: \; p(X=x|Y=1)\, P(Y=1) = P(X=x|Y=0)\, P(Y=0)\}$$

Lets find the decision boundary.

If class distribution is P(Y=1) = Ber(θ) and

class conditional feature distribution P(X=x|Y=y) is d-dim

Gaussian N(μ_y,Σ_y)

$$P(X=x|Y=y) = \frac{1}{\sqrt{(2\pi)^d|\Sigma_y|}} \exp\left(-\frac{(x-\mu_y)^T\Sigma_y^{-1}(x-\mu_y)}{2}\right)$$

Decision Boundary of Gaussian Bayes

- Decision boundary is set of points x: P(Y=1|X=x) = P(Y=0|X=x)

Compute the ratio

$-x^T \Sigma_1^{-1} x$

$+ x^T \Sigma_0^{-1} x$

$$1 = \frac{P(Y=1|X=x)}{P(Y=0|X=x)} = \frac{P(X=x|Y=1)P(Y=1)}{P(X=x|Y=0)P(Y=0)}$$

$$= \sqrt{\frac{|\Sigma_0|}{|\Sigma_1|}} \exp\left(-\frac{(x-\mu_1)^T\Sigma_1^{-1}(x-\mu_1)}{2} + \frac{(x-\mu_0)^T\Sigma_0^{-1}(x-\mu_0)}{2}\right)\frac{\theta}{1-\theta}$$

In general, this implies a quadratic equation in x. But if $\Sigma_1 = \Sigma_0$, then quadratic part cancels out and decision boundary is linear.

Glossary of Machine Learning

- Feature/Attribute
- iid
- Bayes classifier
- Class distribution
- Class conditional distribution of features
- Estimator – hat notation
- MLE
- Decision boundary

Some notes

- Recitation Friday Sept 18

  Recap of MLE/MAP hands on

  Naïve Bayes application

  Linear algebra and multi-variate calculus

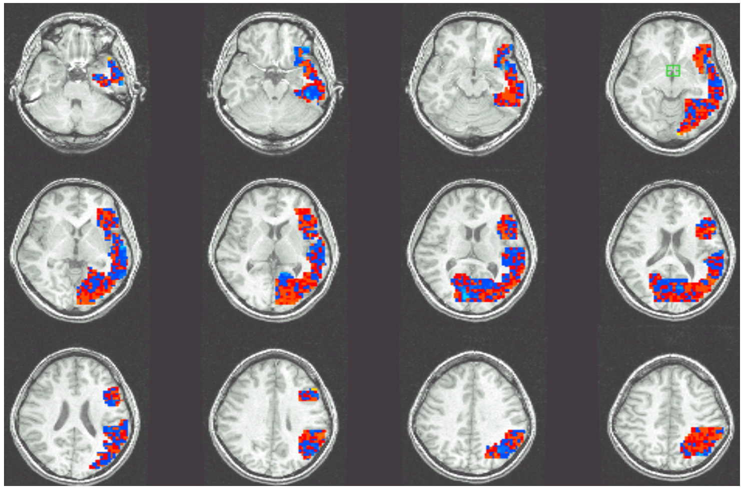
- HW1 due date -> Sept 25

Naïve Bayes
Learning Distributions (MAP)

Aarti Singh

Machine Learning 10-315
Sept 16, 2020

Multi-class, multi-dimensional classification – Continuous features



High Stress
Moderate Stress
Low Stress

Input feature vector, X **Label, Y**

We started with a simple case:

label Y is binary (either “Stress” or “No Stress”)

X is average brain activity in the “Amygdala”

In general: label Y can belong to K>2 classes

X is multi-dimensional d>1 (average activity in all brain regions)

How many parameters do we need to learn (continuous features)?

Class probability: $K=3$

$P(Y = y) = p_y$ for all y in H, M, L $\qquad p_H, p_M, p_L$ (sum to 1)

K-1 if K labels

Class conditional distribution of features:

$\mu_1 \ldots \mu_K$ - Kd

$\Sigma_1 \ldots \Sigma_K$ - $Kd\left(\frac{d+1}{2}\right)$

$P(X=x|Y = y) \sim N(\mu_y,\Sigma_y)$ for each y

μ_y – d-dim vector

Σ_y - dxd matrix

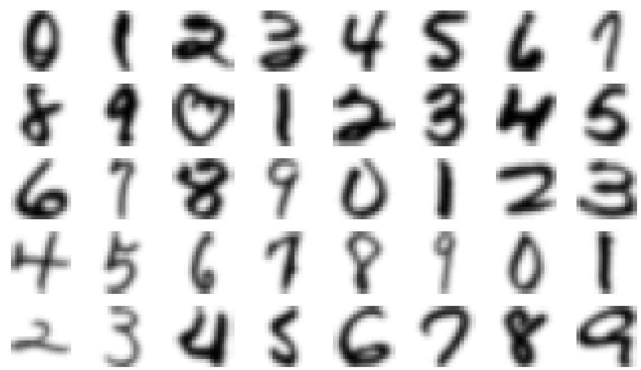
$Kd + Kd(d+1)/2 = O(Kd^2)$ if d features

$3\times(256^2)^2$

Quadratic in dimension d! If d = 256x256 pixels, ~ 13 billion parameters!

Multi-class, multi-dimensional classification - Discrete features

$$X = \begin{bmatrix} x_1 \\ \vdots \\ x_d \end{bmatrix} \quad x_1 \to \{0,1\}$$



Input feature vector, X

"0"
"1"
…
"9"

Label, Y

$$X = \begin{bmatrix} x_1 \\ \vdots \\ x_{|V|} \end{bmatrix}$$

Input feature vector, X

Sports
Science
News

Label, Y

How many parameters do we need to learn (discrete features)?

Class probability: $K=10$

$P(Y = y) = p_y$ for all y in 0, 1, 2, …, 9 $p_0, p_1, \ldots, p_9$ (sum to 1)

K-1 if K labels

Class conditional distribution of (binary) features:

$P(X=x|Y = y)$ ~ For each label y, maintain probability table with 2^d-1 entries

d $\begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \\ \vdots \end{bmatrix}$

$X \in 2^d$

$0, 0, \ldots 0$
$0, 0, \ldots 1$
$0, 0, \ldots 1\,0$
$\vdots$

$X = \begin{bmatrix} x_1 \\ \vdots \\ x_d \end{bmatrix}$

$x_i \in [0,1]$

pixel

$K(2^d – 1)$ if d binary features

Exponential in dimension d!

What's wrong with too many parameters?

- How many training data needed to learn one parameter (bias of a coin)?

- Need lots of training data to learn the parameters!
  - Training data > number of (independent) parameters

runtime + storage requirement

Naïve Bayes Classifier

- Bayes Classifier with additional "naïve" assumption:
  - Features are independent given class:

$$X = \begin{bmatrix} X_1 \\ X_2 \end{bmatrix}$$

$$\begin{aligned} P(X_1, X_2|Y) &= P(X_1|X_2, Y)P(X_2|Y) \leftarrow \text{chain rule} \\ &= P(X_1|Y)P(X_2|Y) \end{aligned}$$

  - More generally:

$$P(X_1 \ldots X_d|Y) = \prod_{i=1}^{d} P(X_i|Y) \qquad X = \begin{bmatrix} X_1 \\ X_2 \\ \ldots \\ X_d \end{bmatrix}$$

- If conditional independence assumption holds, NB is optimal classifier! But worse otherwise.

Conditional Independence

- X is **conditionally independent** of Y given Z:

  probability distribution governing X is independent of the value of Y, given the value of Z

$$(\forall x, y, z) P(X = x | Y = y, Z = z) = P(X = x | Z = z)$$

- Equivalent to:

$$P(X, Y \mid Z) = P(X \mid Z) P(Y \mid Z)$$

- e.g., $P(Thunder | Rain, Lightning) = P(Thunder | Lightning)$

  Note: does NOT mean Thunder is independent of Rain

Conditional vs. Marginal Independence

London taxi drivers: A survey has pointed out a positive and significant correlation between the number of accidents and wearing coats. They concluded that coats could hinder movements of drivers and be the cause of accidents. A new law was prepared to prohibit drivers from wearing coats when driving.

Finally another study pointed out that people wear coats when it rains…

Wearing coats is independent of accidents conditioning on the fact that it rained

Naïve Bayes Classifier

- Bayes Classifier with additional "naïve" assumption:
  – Features are independent given class:

$$\underline{P(X_1 ... X_d|Y)} = \prod_{i=1}^{d} P(X_i|Y)$$

$$\begin{aligned} f_{NB}(\mathbf{x}) &= \arg\max_y \; P(x_1, \ldots, x_d \mid y) P(y) \\ &= \arg\max_y \prod_{i=1}^{d} \underline{P(x_i|y)} P(y) \end{aligned}$$

- How many parameters now?

How many parameters do we need to learn (continuous features)?

Class probability:

$P(Y = y) = p_y$ for all y in H, M, L $\qquad$ p_H, p_M, p_L (sum to 1)

K-1 if K labels

Class conditional distribution of features (using Naïve Bayes assumption):

$$\Sigma^{(y)} = \begin{bmatrix} \sigma_1^{2\,(y)} & & 0 \\ & \sigma_2^{2\,(y)} \ddots & \\ 0 & & \sigma_d^2 \end{bmatrix}$$

$P(X_i = x_i \mid Y = y) \sim N(\mu^{(y)}_i, \sigma^2_i{}^{(y)})$ for each y and each pixel i

2Kd $\qquad$ **if d features**

$$Kd + \frac{Kd(d+1)}{2}$$

Linear instead of Quadratic in dimension d!

How many parameters do we need to learn (discrete features)?

Class probability:

$P(Y = y) = p_y$ for all y in 0, 1, 2, …, 9 $p_0, p_1, \ldots, p_9$ (sum to 1)

K-1 if K labels

Class conditional distribution of (binary) features:

$X = \begin{bmatrix} x_1 \\ \vdots \\ x_d \end{bmatrix}$ $x_1 \to \{0,1\}$

$P(\underline{X_i = x_i} | Y = y)$ – one probability value for each y, pixel i

$P(X = x | Y = y)$
$= P(X_1 = x_1, \ldots X_d = x_d | Y = y)$

Kd if d binary features

Linear instead of Exponential in dimension d!

Naïve Bayes Classifier

- Bayes Classifier with additional "naïve" assumption:
  - Features are independent given class:

$$P(X_1 \ldots X_d | Y) = \prod_{i=1}^{d} P(X_i | Y)$$

$$\begin{aligned} f_{NB}(\mathbf{x}) &= \arg\max_y \; P(x_1, \ldots, x_d \mid y) P(y) \\ &= \arg\max_y \prod_{i=1}^{d} P(x_i | y) P(y) \end{aligned}$$

- Has fewer parameters, and hence requires fewer training data, even though assumption may be violated in practice

Naïve Bayes Algo – Discrete features

- Training Data $\{(X^{(j)}, Y^{(j)})\}_{j=1}^{n}$ $\quad X^{(j)} = (X_1^{(j)}, \ldots, X_d^{(j)})$
- Maximum Likelihood Estimates
  - For Class probability

$$\widehat{P}(y) = \frac{\{\#j : Y^{(j)} = y\}}{n}$$

data with label y

  - For class conditional distribution

$$\widehat{P}(x_i|y) = \frac{\widehat{P}(x_i, y)}{\widehat{P}(y)} = \frac{\{\#j : X_i^{(j)} = x_i, Y^{(j)} = y\}/\cancel{n}}{\{\#j : Y^{(j)} = y\}/\cancel{n}}$$

- NB Prediction for test data $\quad X = (x_1, \ldots, x_d)$

$$Y = \arg\max_y \widehat{P}(y) \prod_{i=1}^{d} \frac{\widehat{P}(x_i, y)}{\widehat{P}(y)}$$

Issues with Naïve Bayes

- **Issue 1:** Usually, features are not conditionally independent:

$$P(X_1 ... X_d | Y) \neq \prod_i P(X_i | Y)$$

  Nonetheless, NB is the single most used classifier particularly when data is limited, works well

- **Issue 2:** Typically use MAP estimates instead of MLE since insufficient data may cause MLE to be zero.

Insufficient data for MLE

- What if you never see a training instance where X_1=a when Y=b?
  - e.g., b={SpamEmail}, a ={‘Earn’}
  - $\widehat{P}(X_1 = a \mid Y = b) = 0$

Spam

Not spam

- Thus, no matter what the values $X_2,\ldots,X_d$ take:

$$\boxed{\widehat{P}(X_1 = a, X_2 \ldots X_d|Y) = \underbrace{\widehat{P}(X_1 = a|Y)}_{0} \prod_{i=2}^{d} \widehat{P}(X_i|Y)} = 0$$

- What now???

Naïve Bayes Algo – Discrete features

- Training Data $\{(X^{(j)}, Y^{(j)})\}_{j=1}^{n}$ $\quad X^{(j)} = (X_1^{(j)}, \ldots, X_d^{(j)})$
- Maximum A Posteriori (MAP) Estimates – add m "virtual" data

Assume priors

Spam $\quad$ Earn $\quad$ Spam

$$Q(Y = b) \qquad Q(X_i = a, Y = b)$$

$$\widehat{P}(X_i = a | Y = b) = \frac{\{\#j : X_i^{(j)} = a, Y^{(j)} = b\} + mQ(X_i = a, Y = b)}{\{\#j : Y^{(j)} = b\} + \underbrace{mQ(Y = b)}_{\#\text{ virtual examples with } Y = b}}$$

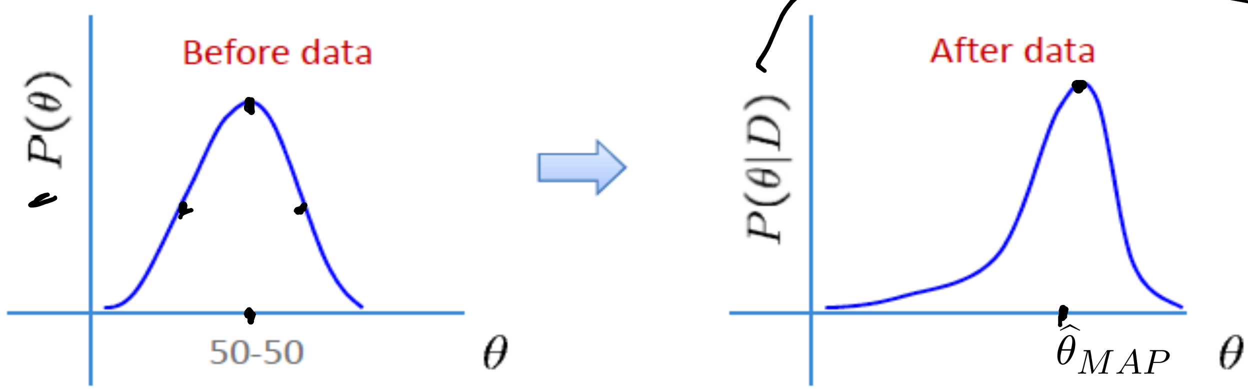
Now, even if you never observe a class/feature posterior probability never zero.

Max A Posteriori (MAP) estimation

Justification for adding virtual examples

- Assume a prior (before seeing data D) distribution P(θ) for parameters θ $X \sim \mathrm{Ber}(\theta)$

posterior



- Choose value that maximizes a posterior distribution P(θ|D) of parameters θ

$$
\begin{aligned}
\widehat{\theta}_{MAP} &= \arg\max_{\theta} \; P(\theta \mid D) \\
&= \arg\max_{\theta} \; P(D \mid \theta) P(\theta)
\end{aligned}
$$

prior

How to choose prior distribution?

- $P(\theta)$
  - Prior knowledge about domain e.g. unbiased coin $P(\theta) = 1/2$
  - A mathematically convenient form e.g. "conjugate" prior

    If $P(\theta)$ is conjugate prior for $P(D|\theta)$, then Posterior has same form as prior

    Posterior ≡ Likelihood x Prior

    $$P(\theta|D) \equiv P(D|\theta) \times P(\theta)$$

    e.g.

Beta	Bernoulli	Beta	θ = bias
Gaussian	Gaussian	Gaussian	θ = mean μ (known Σ)
inv-Wishart	Gaussian	inv-Wishart	θ = cov matrix Σ (known μ)

MAP estimation for Bernoulli r.v.

Choose θ that maximizes a posterior probability

$X \sim Ber(\theta)$
$\in \{0, 1\}$
$\{H, T\}$

$$\begin{aligned}\widehat{\theta}_{MAP} &= \arg\max_{\theta} \quad P(\theta \mid D) \\ &= \arg\max_{\theta} \quad P(\underline{D \mid \theta})P(\theta)\end{aligned}$$

MAP estimate of probability of head (using Beta conjugate prior):

$$P(\theta) \sim Beta(\underline{\beta_H}, \underline{\beta_T})$$

$\theta = P(X=H)$

Beta distribution

$Beta(\beta_H, \beta_T)$

More concentrated as values of β_H, β_T increase