Naïve Bayes Classifier

Manuela Veloso

Co-instructor: Pradeep Ravikumar

Machine Learning 10-701
Feb 6, 2017

Classification

Goal: Construct a **predictor** $f : X \rightarrow Y$ to minimize a risk (performance measure) $R(f)$

Sports
Science
News

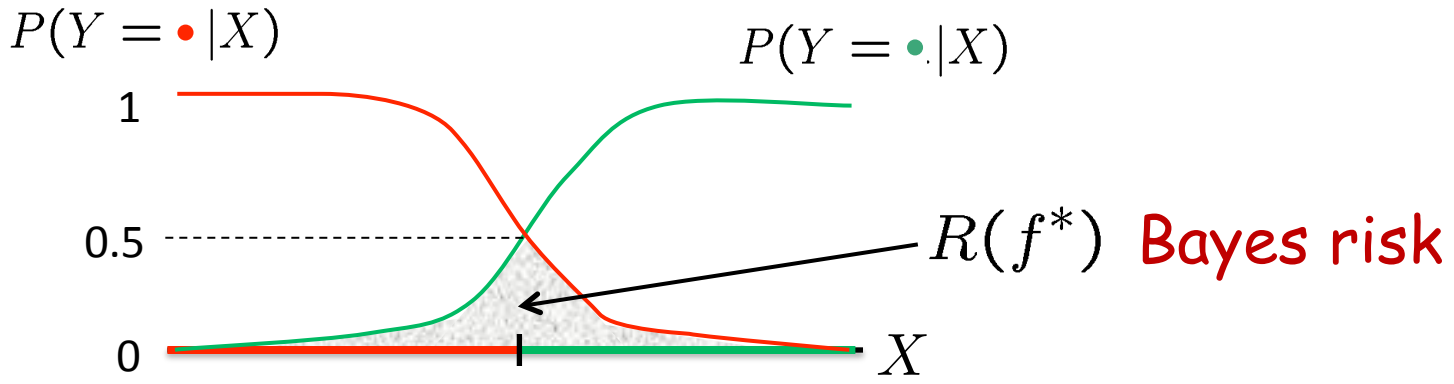
Features, X **Labels, Y**

$$R(f) = P(f(X) \neq Y)$$ **Probability of Error**

Optimal Classification

Optimal predictor: (Bayes classifier)

$$f^* = \arg\min_f P(f(X) \neq Y)$$



$$f^*(x) = \arg\max_{Y=y} P(Y = y|X = x)$$

- Even the optimal classifier makes mistakes R(f*) > 0
- Optimal classifier depends on **unknown** distribution P_{XY}

Optimal Classifier

Bayes Rule: $$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)}$$

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

Optimal classifier:

$$f^*(x) = \arg\max_{Y=y} P(Y = y|X = x)$$

$$= \arg\max_{Y=y} P(X = x|Y = y)P(Y = y)$$

Class conditional density

Class prior

Model based Approach

$$f^*(x) = \arg\max_{Y=y} \underbrace{P(X = x|Y = y)}_{\text{Class conditional distribution}} \underbrace{P(Y = y)}_{\text{Class probability}}$$

We can now consider appropriate models for the two terms

Class probability P(Y=y), Class conditional distribution of features P(X=x|Y=y)

Modeling Class probability P(Y=y) = Bernoulli(θ)

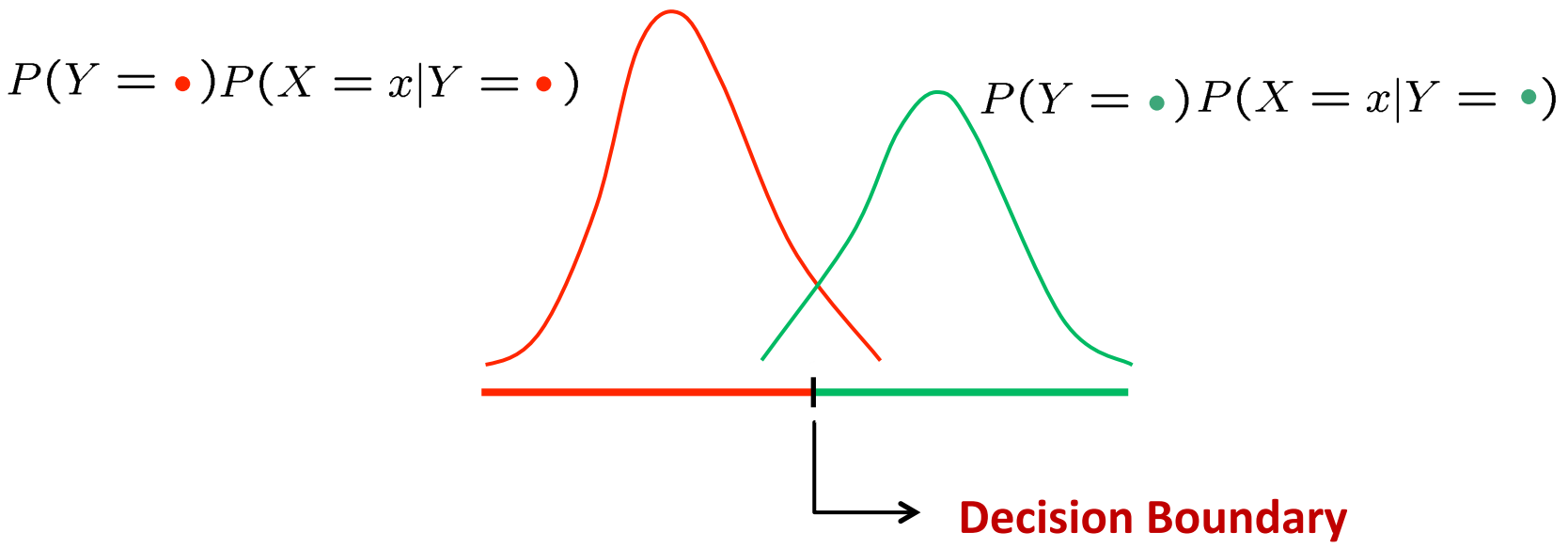
$P(Y = \bullet) = \theta$ $\quad$ $P(Y = \bullet) = 1 - \theta$

Like a coin flip

Modeling Class Conditional Distribution of Features

- Gaussian class conditional densities (1-dimension/feature)

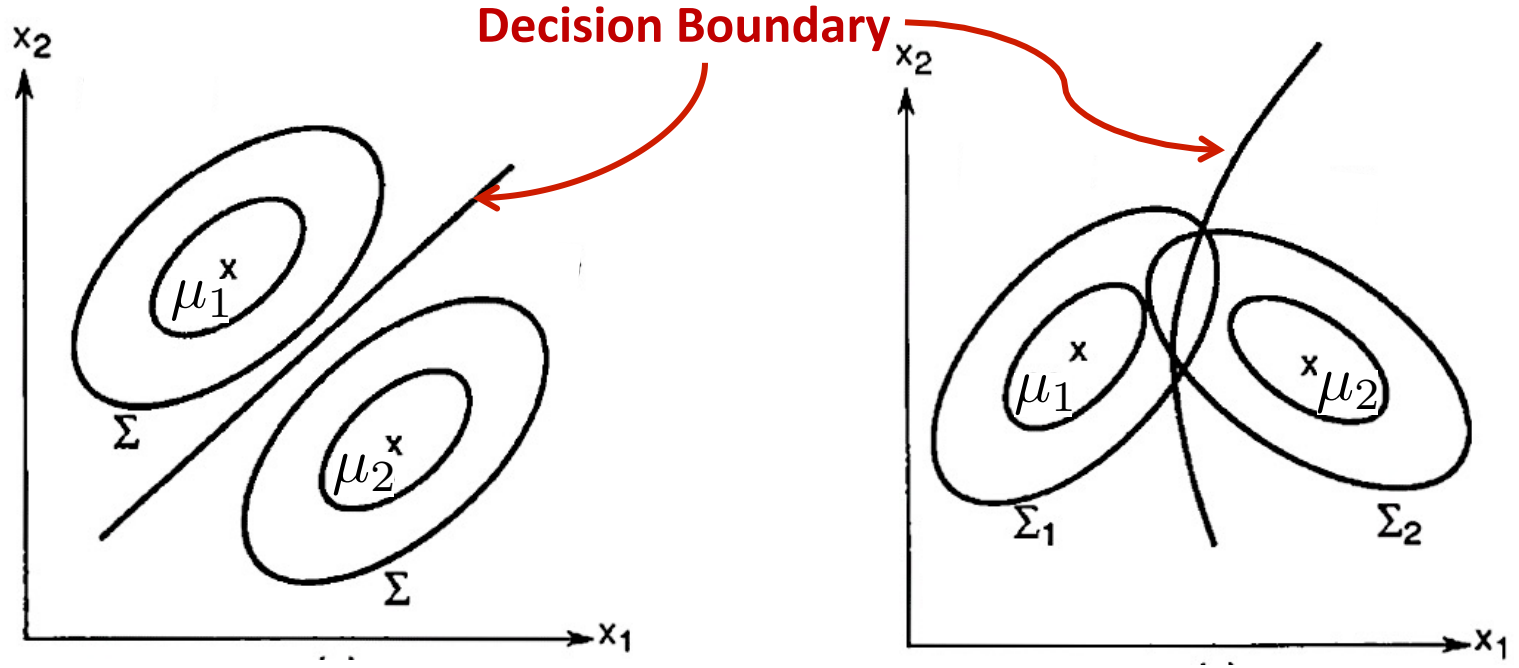
$$P(X = x|Y = y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x-\mu_y)^2}{2\sigma_y^2}\right)$$



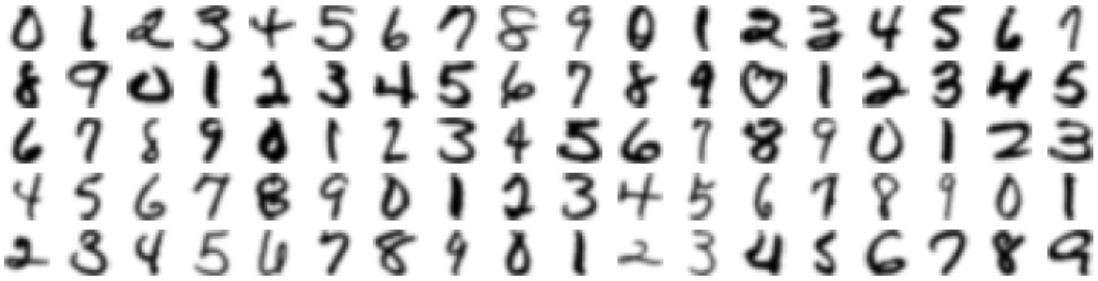
Modeling Class Conditional Distribution of Features

- Gaussian class conditional densities (2-dimensions/features)

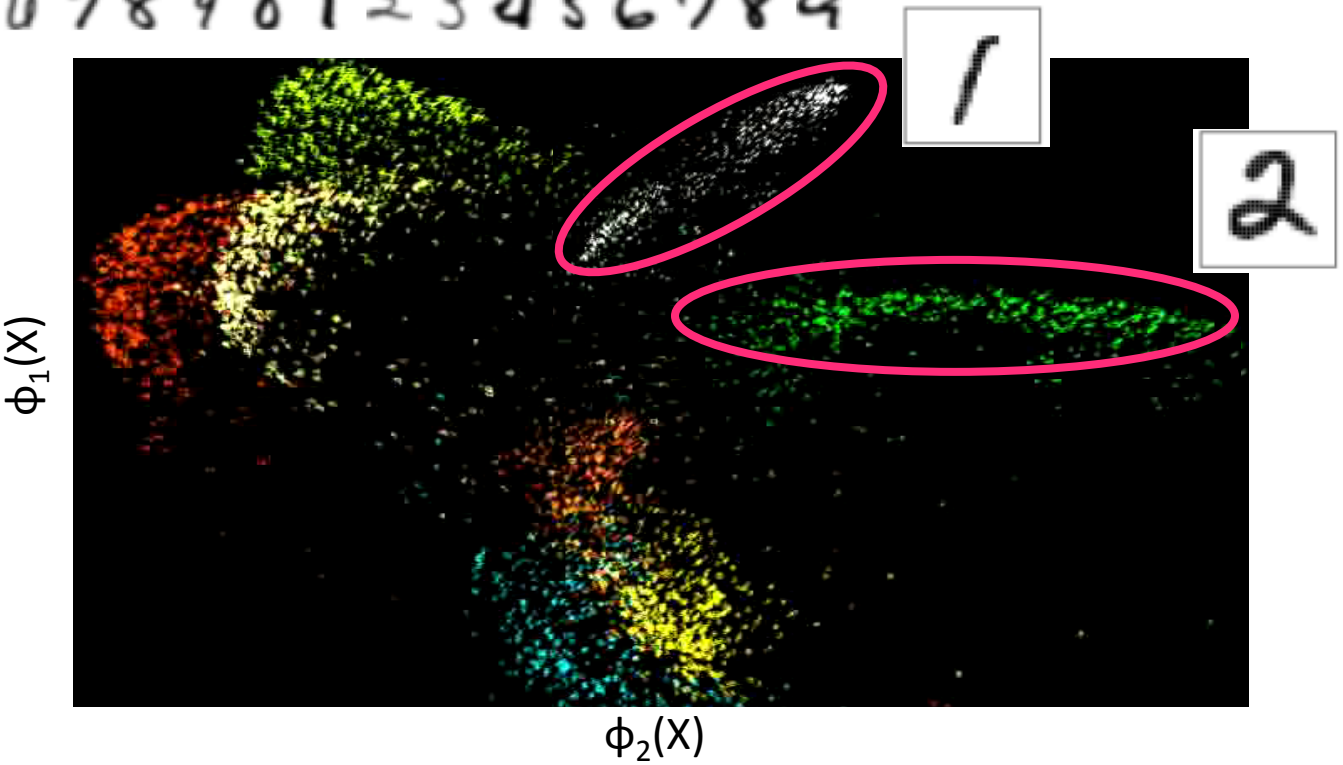
$$P(X = x|Y = y) = \frac{1}{\sqrt{2\pi|\Sigma_y|}} \exp\left(-\frac{(x-\mu_y)\Sigma_y^{-1}(x-\mu_y)'}{2}\right)$$



Handwritten digit recognition



Multi-class classification

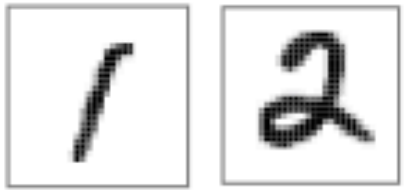


Note: 8 digits shown out of 10 (0, 1, …, 9);
Axes are obtained by nonlinear dimensionality reduction (later in course)

Handwritten digit recognition

Training Data:

Each image represented as a vector of **intensity values at the d pixels (features)**

Input, X  … n greyscale images $X = \begin{bmatrix} X_1 \\ X_2 \\ \dots \\ X_d \end{bmatrix}$

Label, Y 1 2 … n labels

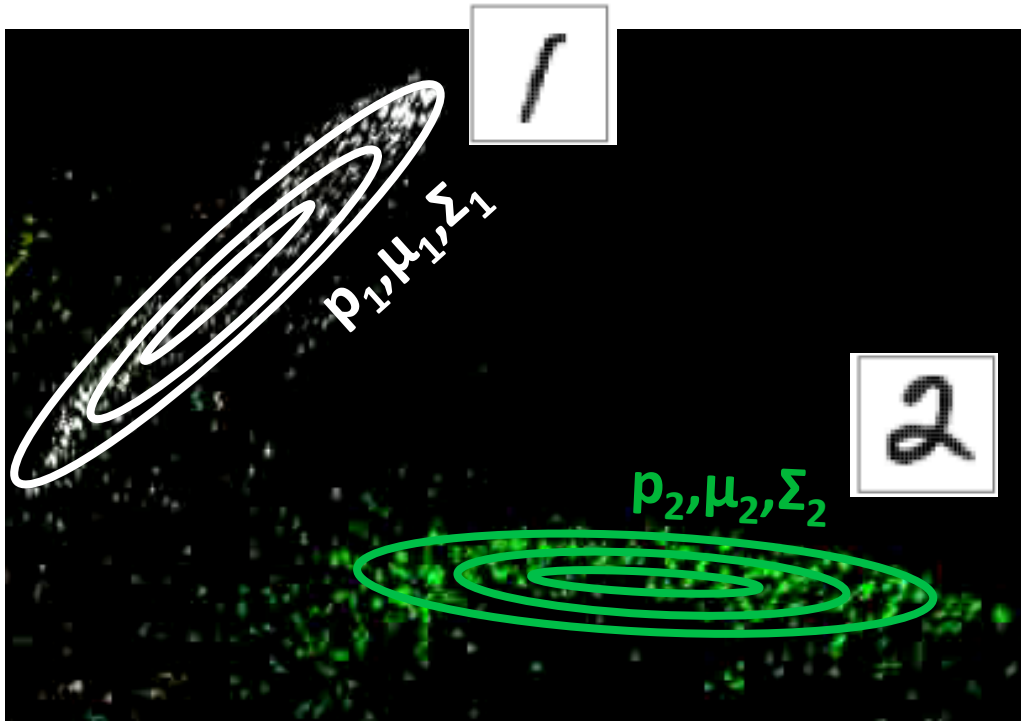
Gaussian Bayes model:

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

$P(X=x|Y = y) \sim N(\mu_y,\Sigma_y)$ for each y μ_y – d-dim vector

Σ_y - dxd matrix

Gaussian Bayes classifier



$P(Y = y) = p_y$ for all y in 0, 1, 2, …, 9

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

$p_0, p_1, \ldots, p_9$ (sum to 1)

μ_y – d-dim vector

Σ_y - dxd matrix

Decision Boundary of Gaussian Bayes

- Binary classification with continuous features

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

If class conditional feature distribution P(X=x|Y=y) is 2-dim Gaussian $N(\mu_y,\Sigma_y)$

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

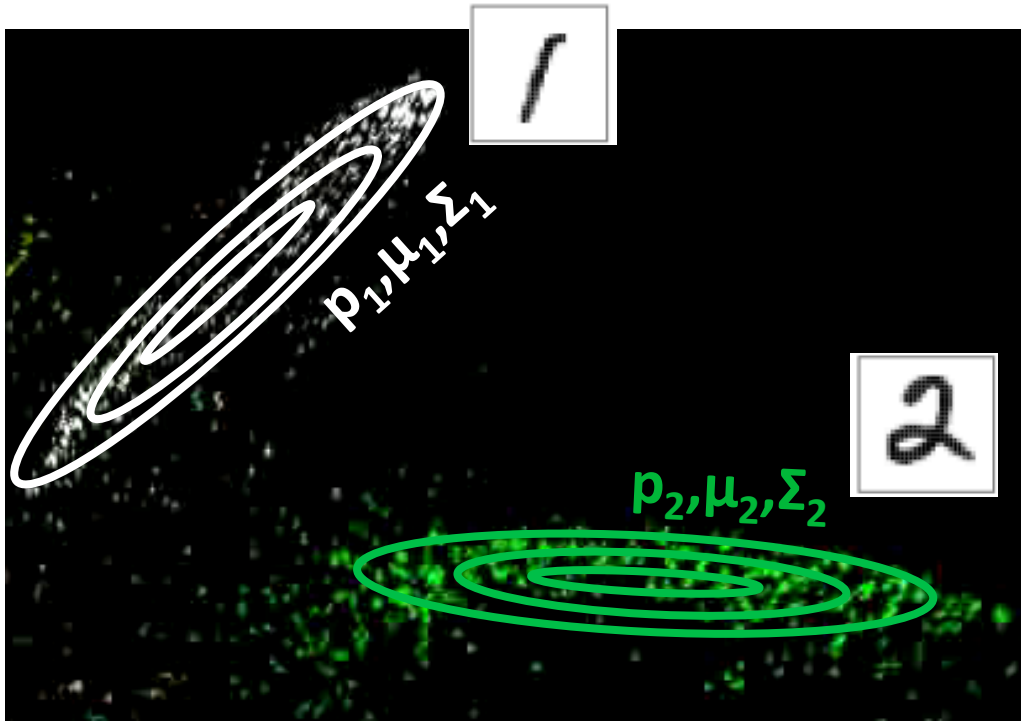
$$\frac{P(Y=1|X=x)}{P(Y=0|X=x)} = \frac{P(X=x|Y=1)P(Y=1)}{P(X=x|Y=0)P(Y=0)}$$

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

Note: In general, this implies a quadratic equation in x.

But if $\Sigma_1 = \Sigma_0$, then quadratic part cancels out and equation is linear.

Gaussian Bayes classifier



How to learn parameters p_y, μ_y, Σ_y from data?

$P(Y = y) = p_y$ for all y in 0, 1, 2, …, 9

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

p_0, p_1, …, p_9 (sum to 1)

μ_y – d-dim vector

Σ_y - dxd matrix

How many parameters do we need to learn?

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 features:

$P(X=x|Y = y) \sim N(\mu_y,\Sigma_y)$ for each y μ_y – d-dim vector

Σ_y - dxd matrix

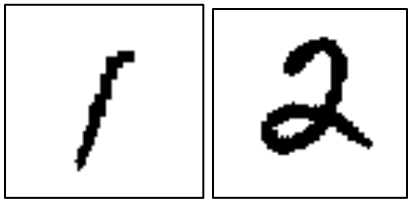
$Kd + Kd(d+1)/2 = O(Kd^2)$ if d features

Quadratic in dimension d! If d = 256x256 pixels, ~ 21.5 billion parameters!

What about discrete features?

Training Data:

Each image represented as a vector of **d binary features (black 1 or white 0)**

Input, X  … n **black-white** images $X = \begin{bmatrix} X_1 \\ X_2 \\ \dots \\ X_d \end{bmatrix}$

Label, Y 1 2 … n labels

Discrete Bayes model:

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

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

How many parameters do we need to learn?

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 features:

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

$K(2^d - 1)$ 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 parameters

Naïve Bayes Classifier

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

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

$$P(X_1, X_2|Y) = P(X_1|X_2, Y)P(X_2|Y)$$
$$= P(X_1|Y)P(X_2|Y)$$

  - More generally:

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

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

- 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:

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

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

- How many parameters now?

Handwritten digit recognition (continuous features)

Training Data:

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

… n greyscale images with d pixels

Y 1 2 … n labels

How many parameters?

Class probability $P(Y = y) = p_y$ for all y **K-1 if K labels**

May not hold

Class conditional distribution of features (using Naïve Bayes assumption)

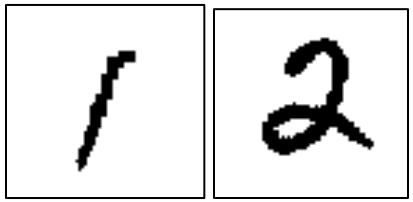
$P(X_i = x_i | Y = y) \sim N(\mu^{(y)}_i, \sigma^{2}_{i}{}^{(y)})$ for each y and each pixel i **2Kd**

Linear instead of Quadratic in d!

Handwritten digit recognition (discrete features)

Training Data:

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



… n black-white (1/0) images with d pixels

Y 1 2 … n labels

How many parameters?

Class probability $P(Y = y) = p_y$ for all y **K-1 if K labels**

May not hold

Class conditional distribution of features (using Naïve Bayes assumption)

$P(X_i = x_i | Y = y)$ – one probability value for each y, pixel i **Kd**

Linear instead of Exponential in d!

Naïve Bayes Classifier

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

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

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

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

Naïve Bayes Algo – Discrete features

- Training Data $\{(X^{(j)}, Y^{(j)})\}_{j=1}^{n}$ $X^{(j)} = (X_1^{(j)}, \ldots, X_d^{(j)})$
- Maximum Likelihood Estimates
  - For Class probability $$\widehat{P}(y) = \frac{\{\#j : Y^{(j)} = y\}}{n}$$
  - For class conditional distribution

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

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

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

Issues with Naïve Bayes

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

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

  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'}
  - $P(X_1 = a \mid Y = b) = 0$
- Thus, no matter what the values $X_2,\ldots,X_d$ take:

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

- What now???

Naïve Bayes Algo – Discrete features

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

Assume priors

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

$$\widehat{P}(X_i = a | Y = b) = \frac{\{\#j : X_i^{(j)} = a, Y^{(j)} = b\} + mQ(X_i = a, Y = b)}{\{\#j : Y^{(j)} = b\} + mQ(Y = b)}$$

virtual examples with Y = b

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

Case Study: Text Classification

- Classify e-mails
  - Y = {Spam,NotSpam}
- Classify news articles
  - Y = {what is the topic of the article?}
- Classify webpages
  - Y = {Student, professor, project, …}

- What about the features **X**?
  - The text!

Bag of words approach

aardvark	0
about	2
all	2
Africa	1
apple	0
anxious	0
...	
gas	1
...	
oil	1
…	
Zaire	0

NB for Text Classification

- Features **X** are the count of how many times each word in the vocabulary appears in document

- Probability table for P(**X**|Y) is huge!!!

- NB assumption helps a lot!!!

- Bag of words + Naïve Bayes assumption imply P(X|Y=y) is just the product of probability of each word, raised to its count, in a document on topic y

$$h_{NB}(\mathbf{x}) = \arg\max_y P(y) \prod_{w=1}^{W} P(w|y)^{count_w}$$

Bag of words model

- Typical additional assumption – **Position in document doesn't matter**
  - "Bag of words" model – order of words on the page ignored
  - Sounds really silly, but often works very well!

When the lecture is over, remember to wake up the person sitting next to you in the lecture room.

Bag of words model

- Typical additional assumption – **Position in document doesn't matter**
  - "Bag of words" model – order of words on the page ignored
  - Sounds really silly, but often works very well!

in is lecture lecture next over person remember room sitting the the the to to up wake when you

NB with Bag of Words for text classification

- Learning phase:
  - Class Prior P(Y): fraction of times topic Y appears in the collection of documents
  - P(w|Y): fraction of times word w appears in documents with topic Y
- Test phase:
  - For each document
    - Use Bag of words + naïve Bayes decision rule

$$h_{NB}(\mathbf{x}) = \arg\max_y P(y) \prod_{w=1}^{W} P(w|y)^{count_w}$$

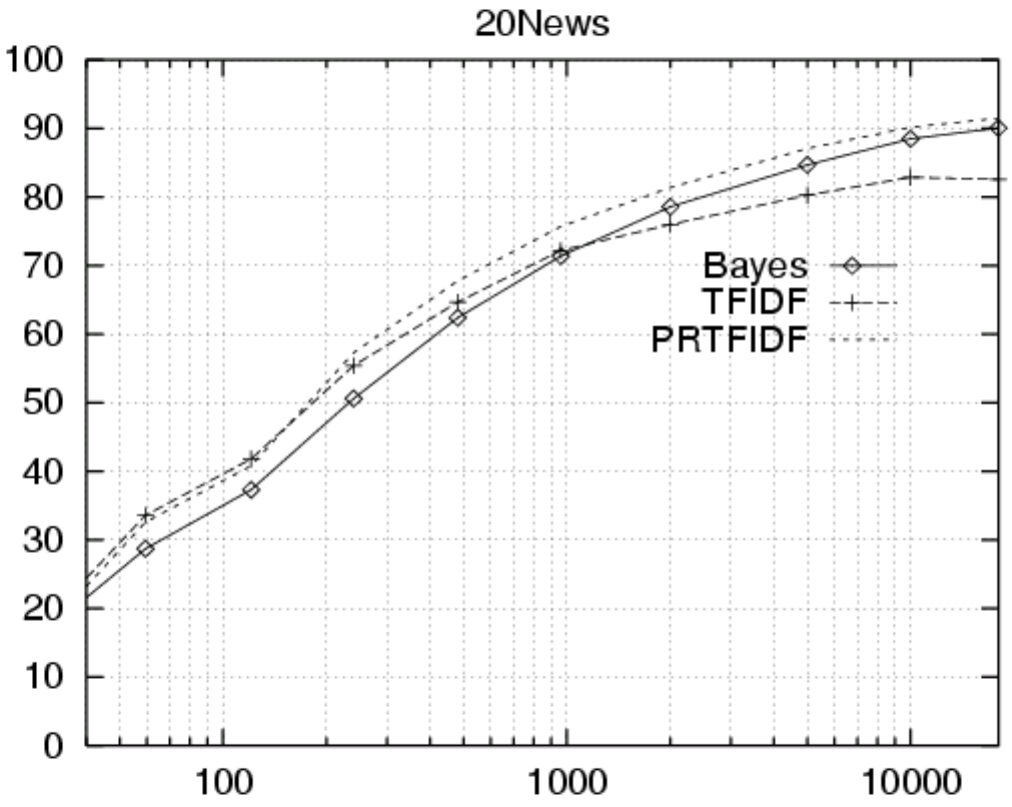
Twenty news groups results

Given 1000 training documents from each group
Learn to classify new documents according to
which newsgroup it came from

comp.graphics	misc.forsale
comp.os.ms-windows.misc	rec.autos
comp.sys.ibm.pc.hardware	rec.motorcycles
comp.sys.mac.hardware	rec.sport.baseball
comp.windows.x	rec.sport.hockey
alt.atheism	sci.space
soc.religion.christian	sci.crypt
talk.religion.misc	sci.electronics
talk.politics.mideast	sci.med
talk.politics.misc	
talk.politics.guns	

Naive Bayes: 89% classification accuracy

Learning curve for twenty news groups



Accuracy vs. Training set size (1/3 withheld for test)

What if features are continuous?

Eg., character recognition: X_i is intensity at ith pixel

Gaussian Naïve Bayes (GNB):

$$P(X_i = x \mid Y = y_k) = \frac{1}{\sigma_{ik}\sqrt{2\pi}} \; e^{\frac{-(x-\mu_{ik})^2}{2\sigma_{ik}^2}}$$

Different mean and variance for each class k and each pixel i.

Sometimes assume variance

- is independent of Y (i.e., σ_i),
- or independent of X_i (i.e., σ_k)
- or both (i.e., σ)

Estimating parameters: Y discrete, X_i continuous

Maximum likelihood estimates:

$$\hat{\mu}_{MLE} = \frac{1}{N}\sum_{j=1}^{N} x_j$$

$$\hat{\mu}_{ik} = \frac{1}{\sum_j \delta(Y^j = y_k)} \sum_j X_i^j \delta(Y^j = y_k)$$

k^{th} class

j^{th} training image

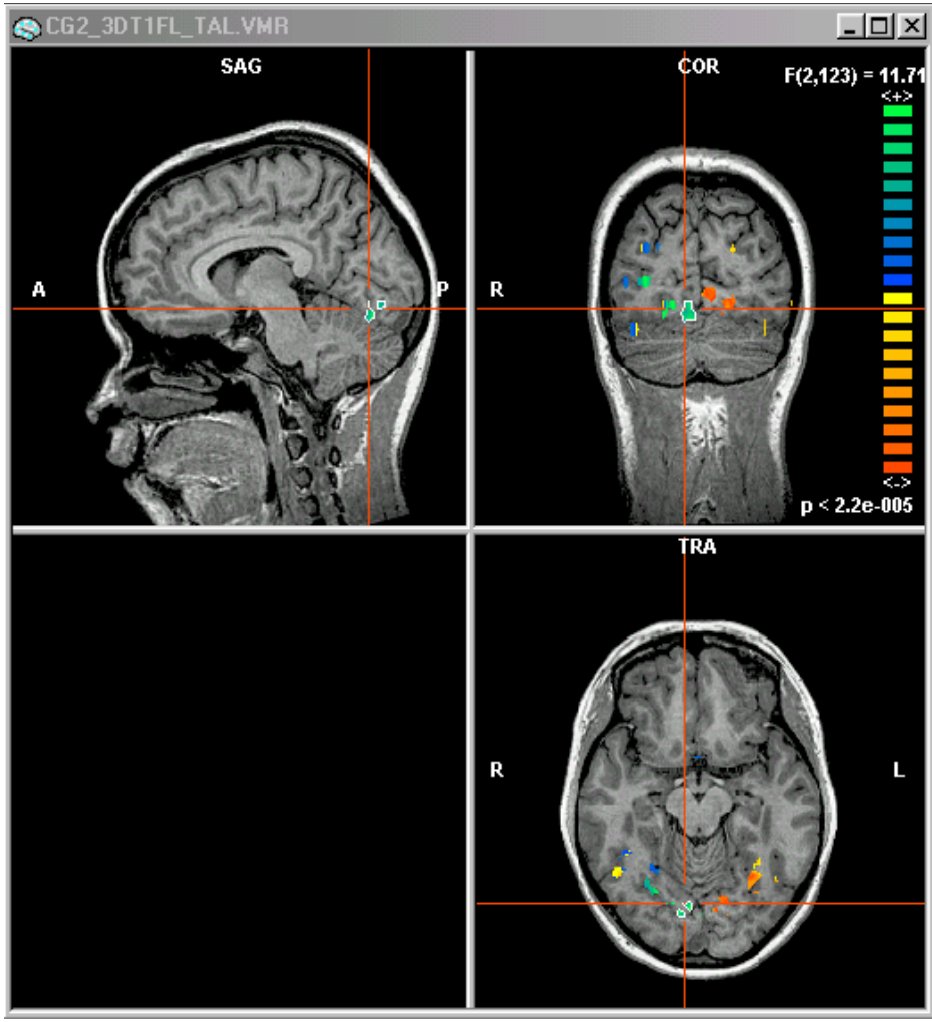
i^{th} pixel in j^{th} training image

$$\hat{\sigma}^2_{unbiased} = \frac{1}{N-1}\sum_{j=1}^{N} (x_j - \hat{\mu})^2$$

$$\hat{\sigma}^2_{ik} = \frac{1}{\sum_j \delta(Y^j = y_k) - 1} \sum_j (X_i^j - \hat{\mu}_{ik})^2 \delta(Y^j = y_k)$$

Example: GNB for classifying mental states

[Mitchell et al.]



~1 mm resolution

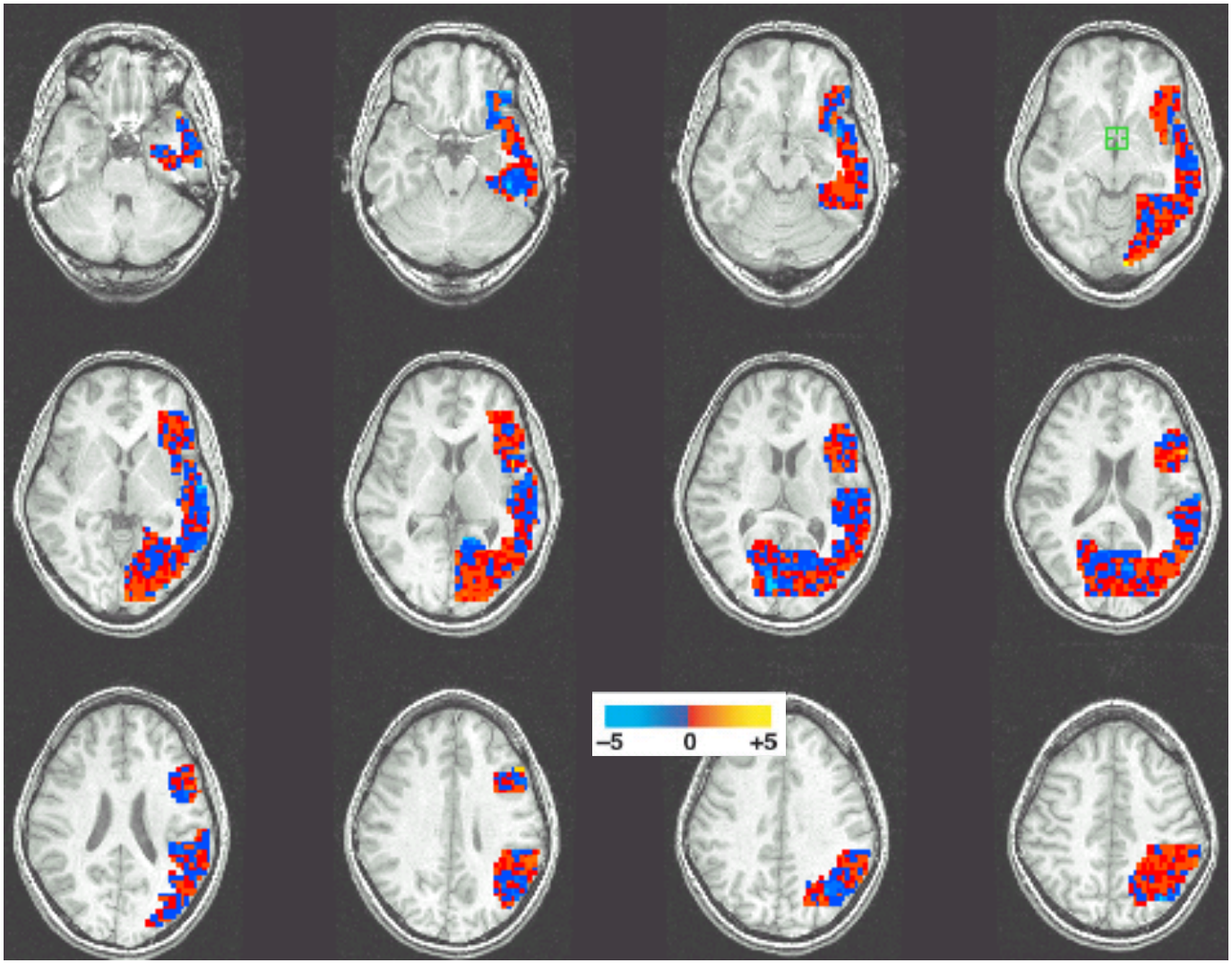
~2 images per sec.

15,000 voxels/image

non-invasive, safe

measures Blood Oxygen Level Dependent (BOLD) response

Gaussian Naïve Bayes: Learned $\mu_{voxel,word}$



[Mitchell et al.]

15,000 voxels or features

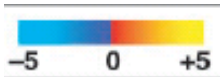
10 training examples or subjects per class (12 word categories)

Learned Naïve Bayes Models –
Means for P(BrainActivity | WordCategory)

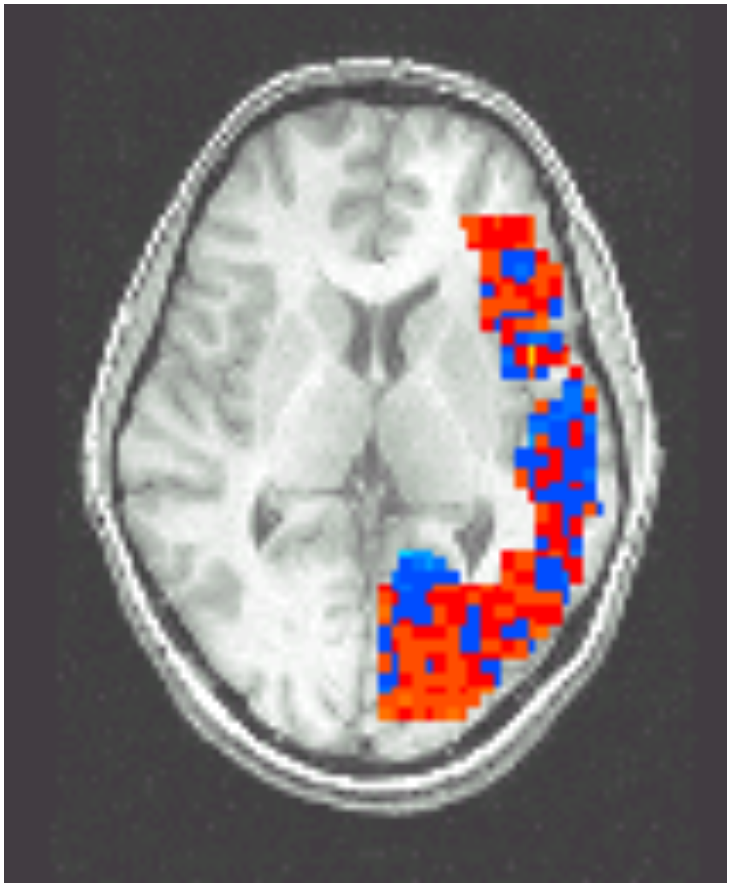
Pairwise classification accuracy: 85%

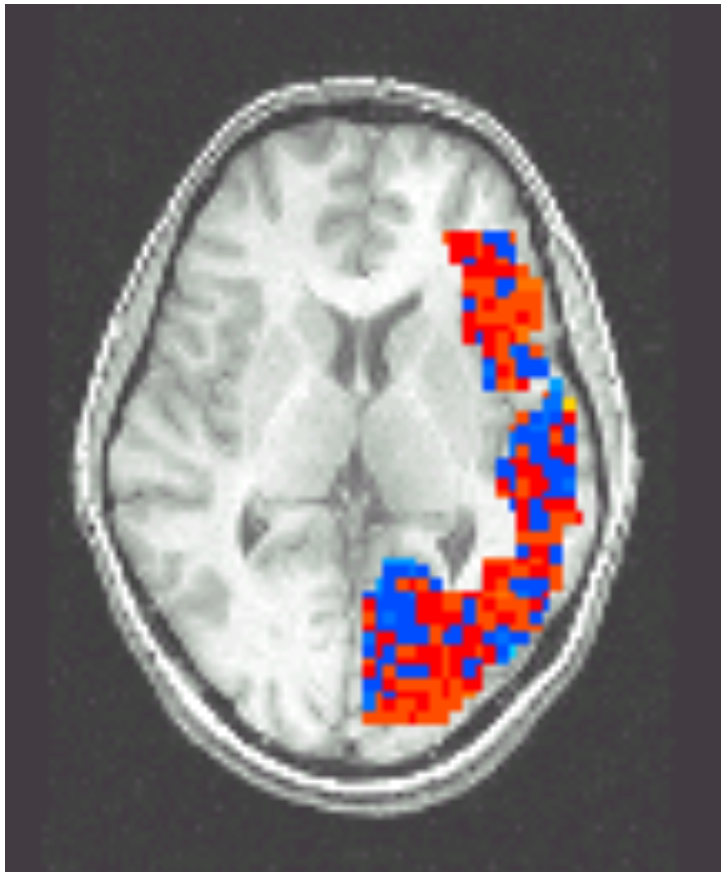
[Mitchell et al.]

People words



Animal words





What you should know…

- Optimal decision using Bayes Classifier
- Naïve Bayes classifier
  - What's the assumption
  - Why we use it
  - How do we learn it
  - Why is MAP estimation important
- Text classification
  - Bag of words model
- Gaussian NB
  - Features are still conditionally independent
  - Each feature has a Gaussian distribution given class

Gaussian Naïve Bayes vs. Logistic Regression

Set of Gaussian Naïve Bayes parameters (feature variance independent of class label)

Set of Logistic Regression parameters

- Representation equivalence (both yield linear decision boundaries)
  - **But only in a special case!!!** (GNB with class-independent variances)
  - **LR makes no assumptions about** P(**X**|Y) **in learning**!!!
  - **Optimize different functions (MLE/MCLE) or (MAP/MCAP)! Obtain different solutions**

Discriminative vs Generative Classifiers

Optimal Classifier:

$$f^*(x) = \arg\max_{Y=y} P(Y = y|X = x)$$
$$= \arg\max_{Y=y} P(X = x|Y = y)P(Y = y)$$

Generative (Model based) approach: e.g. Naïve Bayes

- Assume some probability model for P(Y) and P(X|Y)
- Estimate parameters of probability models from training data

Discriminative (Model free) approach: e.g. Logistic regression

Why not learn P(Y|X) directly? Or better yet, why not learn the decision boundary directly?

- Assume some functional form for P(Y|X) or for the decision boundary
- Estimate parameters of functional form directly from training data

Gaussian Naïve Bayes vs. Logistic Regression

[Ng & Jordan, NIPS 2001]

Given **infinite data** (asymptotically),

If conditional independence assumption holds,
Discriminative LR and generative NB perform similar.

$$\epsilon_{\mathrm{Dis},\infty} \sim \epsilon_{\mathrm{Gen},\infty}$$

If conditional independence assumption does NOT holds,
Discriminative LR outperforms generative NB.

$$\epsilon_{\mathrm{Dis},\infty} < \epsilon_{\mathrm{Gen},\infty}$$

Gaussian Naïve Bayes vs. Logistic Regression

Consider Y boolean, X_i continuous, $X=<X_1 \ldots X_d>$

Number of parameters:

- NB: 4d +1 $\theta, (\mu_{1,y}, \mu_{2,y}, \ldots, \mu_{d,y}), (\sigma^2_{1,y}, \sigma^2_{2,y}, \ldots, \sigma^2_{d,y})$ y = 0,1
  3d +1 if class independent variances
- LR: d+1 $w_0, w_1, \ldots, w_d$

Estimation method:

- NB parameter estimates are uncoupled
- LR parameter estimates are coupled

Gaussian Naïve Bayes vs. Logistic Regression

Given **finite data** (n data points, d features), [Ng & Jordan, NIPS 2001]

$$\epsilon_{\text{Dis},n} \leq \epsilon_{\text{Dis},\infty} + O\left(\sqrt{\frac{d}{n}}\right)$$

$$\epsilon_{\text{Gen},n} \leq \epsilon_{\text{Gen},\infty} + O\left(\sqrt{\frac{\log d}{n}}\right)$$

Naïve Bayes (generative) requires n ~ log d to converge to its asymptotic error, whereas Logistic regression (discriminative) requires n ~ d.

Why? "Independent class conditional densities"

* parameter estimates not coupled – each parameter is learnt independently, not jointly, from training data.

Naïve Bayes vs Logistic Regression

Verdict

Both learn a linear boundary (assuming class-ind feature variance)

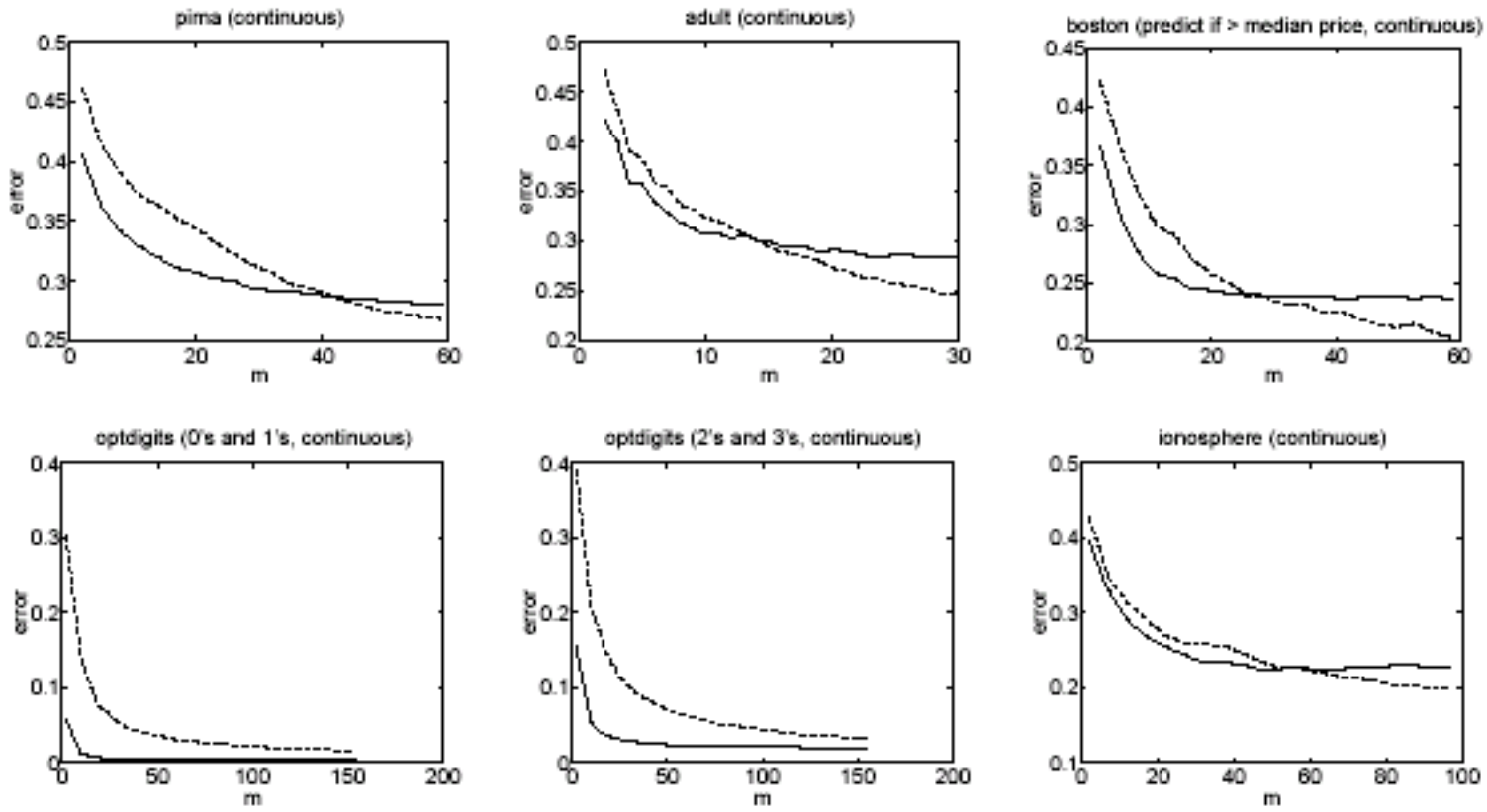
Naïve Bayes makes more restrictive assumptions and has higher asymptotic error,

BUT

converges faster to its less accurate asymptotic error.

Experimental Comparison (Ng-Jordan'01)

UCI Machine Learning Repository 15 datasets, 8 continuous features, 7 discrete features



More in Paper…

—— Naïve Bayes ----- Logistic Regression