Naive Bayes

GU 4241/GR 5241

Statistical Machine Learning

Xiaofei Shi

Tasks

Input → Regressor → Predict real number

Input → Classifier → Predict category

Input → Density Estimator → Probability

Types of classifiers

- Discriminative classifiers:
  - Directly estimate a decision rule/boundary
  - e.g. decision tree, SVM
- Instance based classifiers:
  - Use observation directly
  - e.g. K nearest neighborhood
- Generative classifiers:
  - Build a generative statistical model
  - e.g. Bayesian Network

Classification

- Goal: Construct a predictor

  f: X→ Y   to minimize the risk   R(f)
- R(f) is a performance measure
- In discriminative classifiers we use

  Probability of error

  R(f) = 1- P[f(X) = Y]

Features, X

Sports
Science
News

Labels, Y

COLUMBIA UNIVERSITY

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

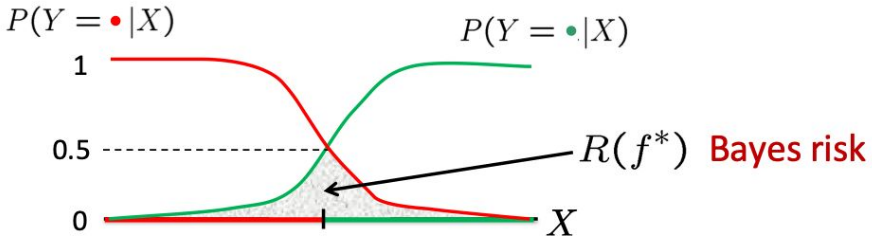
Why not learn P(Y|X) directly? Or better yet, why not learn the decision boundary directly?

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

Optimal Classification

Optimal predictor:
(Bayes classifier)

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



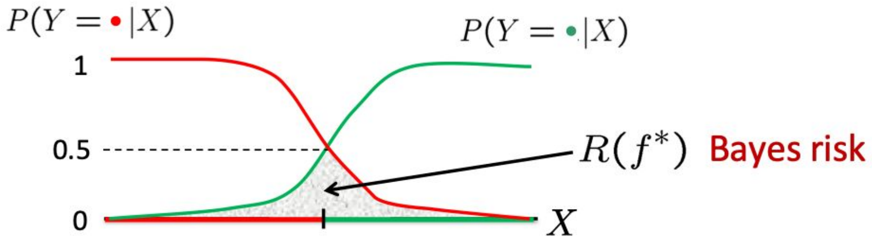
$$f^*(x) = \arg\max_{Y=y} P(Y = y|X = x)$$

Optimal Classification

- Even the best classifier is allowed to make mistakes, i.e. R(f*) > 0.
- Optimal classifier depends on the unknown joint distribution P(X,Y)

Optimal predictor:
(Bayes classifier)

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



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

COLUMBIA UNIVERSITY

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:

$$\begin{aligned} f^*(x) &= \arg\max_{Y=y} P(Y = y|X = x) \\ &= \arg\max_{Y=y} \underbrace{P(X = x|Y = y)}_{\text{Class conditional density}} \underbrace{P(Y = y)}_{\text{Class prior}} \end{aligned}$$

COLUMBIA UNIVERSITY

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 therefore consider appropriate models for these 2 terms

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

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

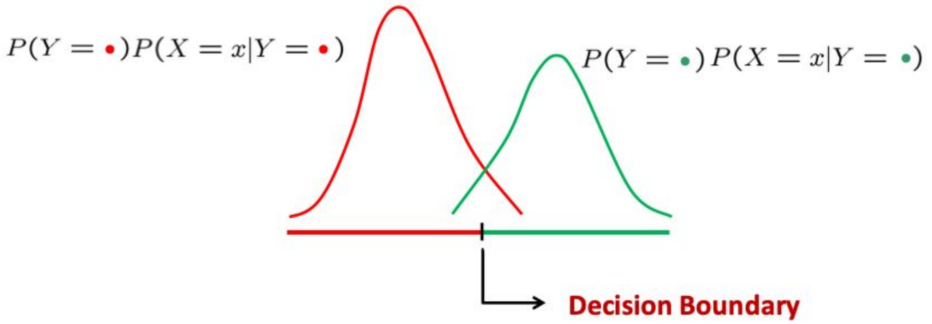
Like a coin flip

Columbia University

More complicated models...

- We model the class conditional distribution of features
- One popular choice is Gaussian class conditional density (1-dim/feature)

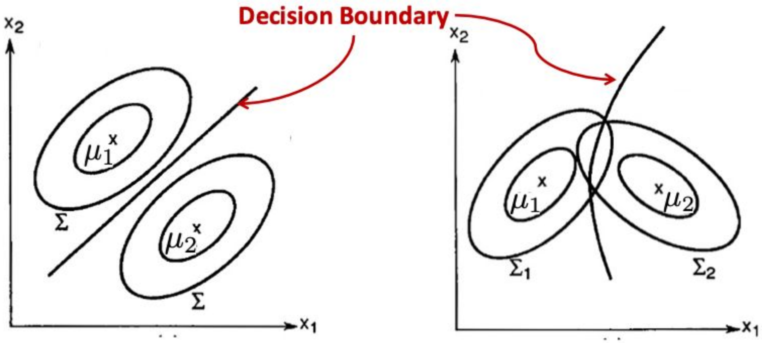
$$P(X = x|Y = y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x-\mu_y)^2}{2\sigma_y^2}\right)$$



More complicated models...

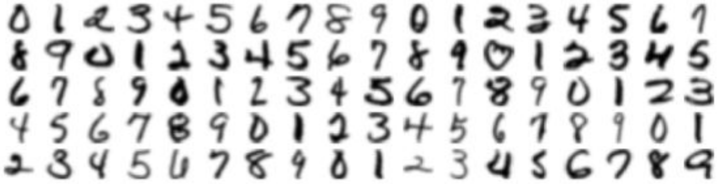
- We model the class conditional distribution of features
- One popular choice is Gaussian class conditional density (1-dim/feature)

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

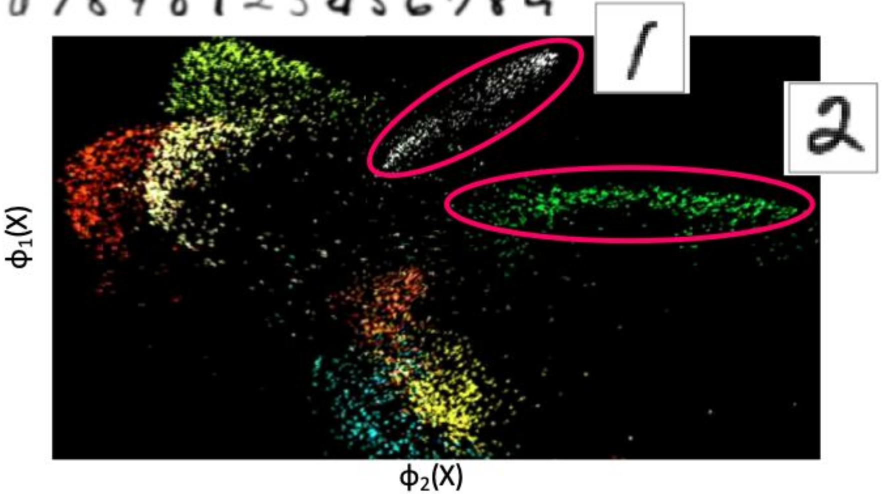


COLUMBIA UNIVERSITY

Handwritten digit recognition (MNIST)



Multi-class classification

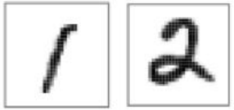


Handwritten digit recognition (MNIST)

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

$p_0, p_1, \ldots, p_9$ (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

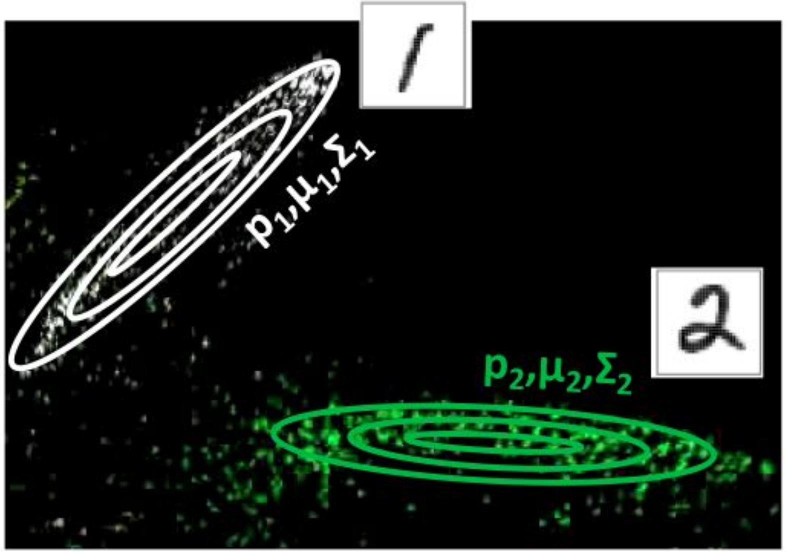
$p_0, p_1, \ldots, p_9$ (sum to 1)

μ_y – d-dim vector

Σ_y - dxd matrix

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

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



Decision boundary of Gaussian Bayes

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

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

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

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

- In general, this implies a quadratic equation in x.
- But in some special cases the quadratic part cancels out and hence the boundary is linear.

Columbia University

Gaussian Bayes Classifier

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

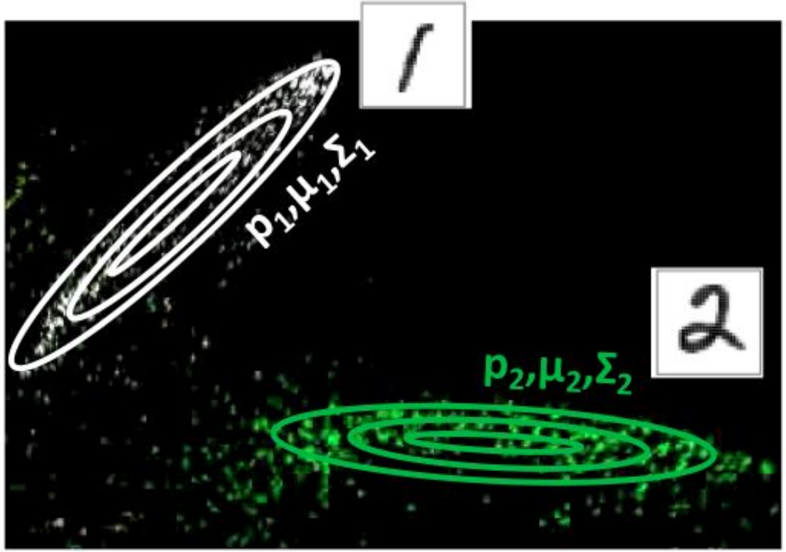
$p_0, p_1, \ldots, p_9$ (sum to 1)

μ_y – d-dim vector

Σ_y - dxd matrix

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

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



How many parameters do we have to learn?

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 have 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)

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

Columbia University

How many parameters do we have to learn?

Class probability:

$P(Y = y) = p_y$ for all y in 0, 1, 2, ..., 9 $\quad$ $p_0, p_1, \ldots, p_9$ (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 $\quad$ 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!

Naive Bayes Classifier

If conditional independence holds, Naive Bayes classifier is the best 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:

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

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

Columbia University

Recall conditional independence

- X is conditionally independent of Y given Z:

  P[X = x| Y=y, Z=z] = P[X = x|Z=z]

- Or equivalently,

  P[X=x, Y=y|Z=z] = P[X=x|Z=z] P[Y=y|Z=z]

$$P(Thunder|Rain, Lightning) = P(Thunder|Lightning)$$

Note: does NOT mean Thunder is independent of Rain

COLUMBIA UNIVERSITY

Naive Bayes Classifier

- Bayes classifier with additional naive assumption:

  - features are independent given class:

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

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

- How many parameters we have now?

Naive Bayes Classifier

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)

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

Naive Bayes Classifier

- Bayes classifier with additional naive assumption:

  - features are independent given class:

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

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

- Fewer parameters and hence requires fewer training data, even though the conditional independence assumptions might be violated in practice.

Naive Bayes Classifier - Algorithm

- 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
$$\widehat{P}(y) = \frac{\{\#j : Y^{(j)} = y\}}{n}$$
  - 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 $X = (x_1, \ldots, x_d)$
$$Y = \arg\max_y \widehat{P}(y) \prod_{i=1}^{d} \frac{\widehat{P}(x_i, y)}{\widehat{P}(y)}$$

COLUMBIA UNIVERSITY

Issues with Naive Bayes

- **<u>Issue 1:</u>** 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

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

COLUMBIA UNIVERSITY

Naive Bayes Classifier - Algorithm

- 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” datapts

Assume given some prior distribution (typically uniform):

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

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

COLUMBIA UNIVERSITY

What if the features are continuous

Gaussian Naïve Bayes (GNB):

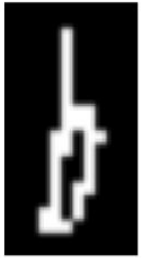
$$P(X_i = x \mid Y = y_k) = \frac{1}{\sigma_{ik}\sqrt{2\pi}} \; e^{\frac{-(x-\mu_{ik})^2}{2\sigma_{ik}^2}}$$

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)

character recognition: X_i is intensity at ith pixel



Takeaways

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

References

- Christopher Bishop: Pattern Recognition and Machine Learning, Chapter 4
- Ziv Bar-Joseph, Tom Mitchell, Pradeep Ravikumar and Aarti Singh: CMU 10-701

Columbia University