CS 4602

Introduction to Machine Learning

Bayesian Classifiers

Instructor: Po-Chih Kuo

Announcement

- Homework 1 due this Thu (10/14)
- Homework 2 will be given on next Thu

Roadmap

- Introduction and Basic Concepts
- Regression (Error-Based Learning)
- Bayesian Classifiers (Probability-Based Learning)
- Decision Trees (Information-Based Learning)
- KNN (Similarity-Based Learning)
- Linear Classifier
- Neural Networks
- Deep learning
- Convolutional Neural Networks
- RNN/Transformer
- Reinforcement Learning
- Model Selection and Evaluation
- Clustering
- Data Exploration & Dimensionality reduction

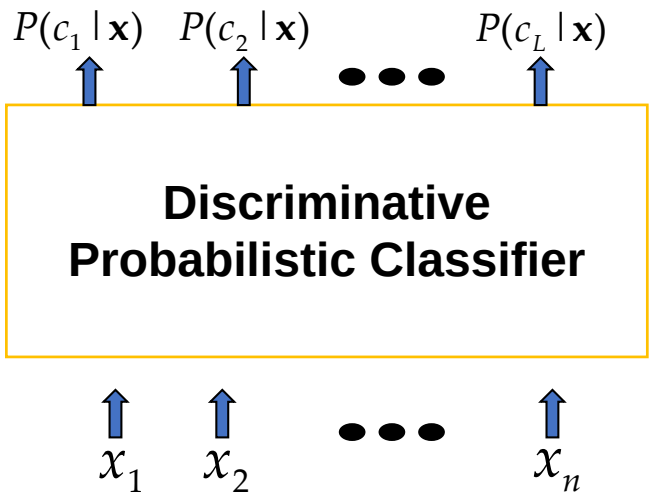
Bayesian Classification

- Statistical method for classification
- Supervised learning approach
- Assumes an underlying probabilistic model, the Bayes theorem
- Can solve problems involving both categorical and continuous valued attributes

Probabilistic Classification

- Discriminative model

$$P(C|\mathbf{X}),\ C = c_1,\cdots,c_L,\mathbf{X} = (X_1,\cdots,X_n)$$



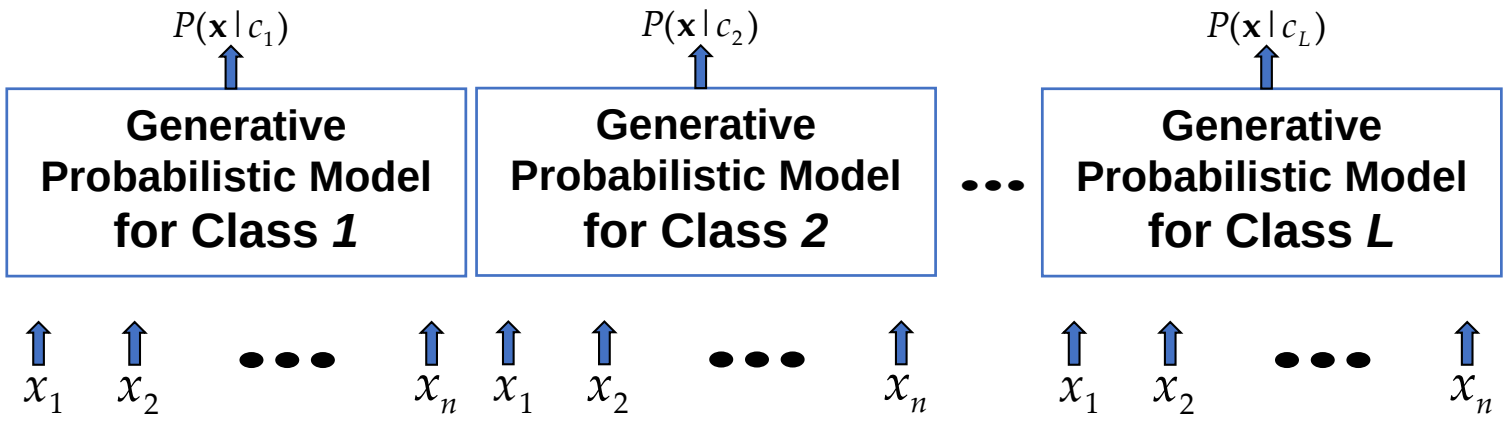
Model a classification rule directly!

Examples: k-NN, decision trees, perceptron, SVM

Probabilistic Classification

- Generative model

$$P(\boldsymbol{X}|C) \rightarrow P(C|\mathbf{X}),\ C = c_1,\cdots,c_L, \mathbf{X} = (X_1,\cdots,X_n)$$

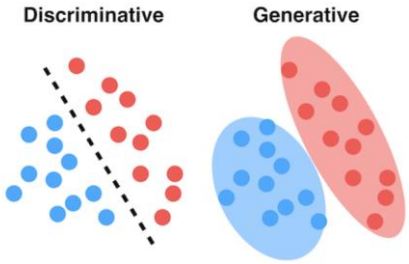


$$\mathbf{x} = (x_1, x_2, \cdots, x_n)$$

Make a probabilistic model of data within each class

Examples: ***naive Bayes***, model-based classifiers

Discriminative vs. Generative



- A generative model learns the **joint** probability distribution p(x,y)
- A discriminative model learns the **conditional** probability distribution p(y|x)
- A simple example: Suppose you have the following data in the form (x,y):

  (1,0), (1,0), (2,0), (2, 1)

  p(x,y) is

	Y=0	Y=1
X=1	1/2	0
X=2	1/4	1/4

  p(y|x) is

	Y=0	Y=1
X=1	1	0
X=2	1/2	1/2

- The distribution p(y|x) is the natural distribution for classifying a given example x into a class y
- p(x,y) can be transformed into p(y|x) by applying **Bayes rule** and then used for classification.
- The distribution p(x,y) can also be used for other purposes. For example, you could use p(x,y) to *generate* likely (x,y) pairs.

The Bayes theorem

X: 35 yrs old with an income of $40,000 and fair credit rating.
C: Hypothesis that the customer will buy a car

The Bayes theorem

- The Bayes Theorem:
  - $P(C|X) = \frac{P(X|C)P(C)}{P(X)}$

$$Posterior = \frac{Likelihood \times Prior}{Evidence}$$

- P(C|X) : Posterior
  - Probability that the customer will buy a car given that we know his age, credit rating and income.
- P(C) : Prior
  - Probability that the customer will buy a car regardless of age, credit rating, income.
- P(X|C) : Likelihood
  - Probability that the customer is 35 yrs old, have fair credit rating and earns $40,000, given that he has bought a car
- P(X) : Evidence
  - Probability that a person from our set of customers is 35 yrs old, have fair credit rating and earns $40,000.

Classification with MAP

- MAP classification rule (Discriminative model)
  - **MAP**: **M**aximum **A** **P**osterior
  - Assign $\boldsymbol{x}$ to c^* if

$$P(C = c^*|\mathbf{X} = \mathbf{x}) > P(C = c|\mathbf{X} = \mathbf{x}) \quad c \neq c^*,\ c = c_1,\cdots,c_L$$

- Generative classification with the MAP rule
  1. Apply Bayesian rule to convert them into posterior probabilities

$$P(C = c_i|\mathbf{X} = \mathbf{x}) = \frac{P(\mathbf{X} = \mathbf{x}|C = c_i)P(C = c_i)}{P(\mathbf{X} = \mathbf{x})}$$
$$\propto P(\mathbf{X} = \mathbf{x}|C = c_i)P(C = c_i), \text{for } i = 1,2,\cdots,L$$

  2. Then apply the MAP rule

Naïve Bayes

- Bayes classification

$$P(C \mid \mathbf{X}) \propto P(\mathbf{X} \mid C)P(C) = P(X_1, \cdots, X_n \mid C)P(C)$$

Difficulty: learning the joint probability $P(X_1, \cdots, X_n \mid C)$

- Naïve Bayes classification
  - Assumption that all input attributes are conditionally independent!

$$\begin{aligned} P(X_1, X_2, \cdots, X_n|C) &= P(X_1|X_2, \cdots, X_n, C)P(X_2, \cdots, X_n|C) \\ &= P(X_1|C)P(X_2, \cdots, X_n|C) \\ &= P(X_1|C)P(X_2|C) \cdots P(X_n|C) \end{aligned}$$

Product of individual probabilities

  - MAP classification rule: for $\mathbf{x} = (x_1, x_2, \cdots, x_n)$

$$[P(x_1|c^*) \cdots P(x_n|c^*)]P(c^*) > [P(x_1|c) \cdots P(x_n|c)]P(c),\ c \neq c^*, c = c_1, \cdots, c_L$$

Naïve Bayes

- Naïve Bayes Algorithm (for discrete input attributes) has two phases

1. Learning Phase: Given a training set **S**,

For each target value of c_i $(c_i = c_1, \cdots, c_L)$
$\hat{P}(C = c_i) \leftarrow$ estimate $P(C = c_i)$ with examples in **S**;
For every attribute value x_{jk} of each attribute X_j $(j = 1, \cdots, n;\ k = 1, \cdots, N_j)$
$\hat{P}(X_j = x_{jk} | C = c_i) \leftarrow$ estimate $P(X_j = x_{jk} | C = c_i)$ with examples in **S**;

Output: conditional probability tables; for $X_j, N_j \times L$ elements

2. Test Phase: Given an unknown instance $\mathbf{X}' = (a'_1, \cdots, a'_n)$,

Look up tables to assign the label c^* to **X'** if

$$[\hat{P}(a'_1|c^*) \cdots \hat{P}(a'_n|c^*)]\hat{P}(c^*) > [\hat{P}(a'_1|c) \cdots \hat{P}(a'_n|c)]\hat{P}(c), \qquad c \neq c^*, c = c_1, \cdots, c_L$$

Example 1

Compute all probabilities required for classification

$$P(C=t)=\frac{1}{2} \qquad P(C=f)=\frac{1}{2}$$

$$P(A=m \mid C=t)=\frac{2}{5} \qquad P(A=g \mid C=t)=\frac{2}{5} \qquad P(A=h \mid C=t)=\frac{1}{5}$$

$$P(A=m \mid C=f)=\frac{1}{5} \qquad P(A=g \mid C=f)=\frac{1}{5} \qquad P(A=h \mid C=f)=\frac{2}{5}$$

$$P(B=b \mid C=t)=\frac{1}{5} \qquad P(B=s \mid C=t)=\frac{1}{5} \qquad P(B=q \mid C=t)=\frac{2}{5}$$

$$P(B=b \mid C=f)=\frac{2}{5} \qquad P(B=s \mid C=f)=\frac{2}{5} \qquad P(B=q \mid C=f)=\frac{2}{5}$$

Training

A	B	C
m	b	t
m	s	t
g	q	t
h	s	t
g	q	t
g	q	f
g	s	f
h	b	f
h	q	f
m	b	f

A = m, B=q, C=?

Test

- For class C = t, we have

$$\Pr(C=t)\prod_{j=1}^{2}\Pr(A_j=a_j \mid C=t)=\frac{1}{2}\times\frac{2}{5}\times\frac{2}{5}=\frac{2}{25}$$

- For class C = f, we have

$$\Pr(C=f)\prod_{j=1}^{2}\Pr(A_j=a_j \mid C=f)=\frac{1}{2}\times\frac{1}{5}\times\frac{2}{5}=\frac{1}{25}$$

- C = t is more probable. t is the final class.

Example 2

- Example: Play Tennis

PlayTennis: training examples

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

PlayTennis: training examples

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

The **learning phase**

$P(\text{Play}=Yes) = 9/14$

$P(\text{Play}=No) = 5/14$

We have four variables, we calculate for each

Outlook	Play=*Yes*	Play=*No*
Sunny	2/9	3/5
Overcast	4/9	0/5
Rain	3/9	2/5

Temperature	Play=*Yes*	Play=*No*
Hot	2/9	2/5
Mild	4/9	2/5
Cool	3/9	1/5

Humidity	Play=*Yes*	Play=*No*
High	3/9	4/5
Normal	6/9	1/5

Wind	Play=*Yes*	Play=*No*
Strong	3/9	3/5
Weak	6/9	2/5

The **testing phase**

- Given a new instance of variable values,
  $\mathbf{x}'$=(Outlook=*Sunny*, Temperature=*Cool*, Humidity=*High*, Wind=*Strong*)
- Given calculated Look up tables

P(Outlook=*Sunny*|Play=*Yes*) = 2/9
P(Temperature=*Cool*|Play=*Yes*) = 3/9
P(Huminity=*High*|Play=*Yes*) = 3/9
P(Wind=*Strong*|Play=*Yes*) = 3/9
P(Play=*Yes*) = 9/14

P(Outlook=S*unny*|Play=*No*) = 3/5
P(Temperature=*Cool*|Play==*No*) = 1/5
P(Huminity=*High*|Play=*No*) = 4/5
P(Wind=*Strong*|Play=*No*) = 3/5
P(Play=*No*) = 5/14

- Use the MAP rule to calculate Yes or No

P(*Yes*|$\mathbf{x}'$): [P(*Sunny*|Yes)P(*Cool*|*Yes*)P(*High*|Yes)P(*Strong*|*Yes*)]P(Play=*Yes*) = 0.0053
P(*No*|$\mathbf{x}'$): [P(*Sunny*|No) P(*Cool*|No)P(*High*|*No*)P(*Strong*|*No*)]P(Play=*No*) = 0.0206

Given the fact P(*No*|x′) > P(*Yes*|x′), we label x′ to be "*No*".

Issues Relevant to Naïve Bayes

1. Violation of independence assumption

   Events are usually correlated

   - For many real world tasks, $P(X_1,\cdots,X_n \mid C) \neq P(X_1 \mid C)\cdots P(X_n \mid C)$
   - Nevertheless, naïve Bayes works surprisingly well anyway!

2. Zero conditional probability problem

   – Such problem exists when no example contains the attribute value $X_j = a_{jk}, \hat{P}(X_j = a_{jk} \mid C = c_i) = 0$

   – In this circumstance, $\hat{P}(x_1 \mid c_i)\cdots\hat{P}(a_{jk} \mid c_i)\cdots\hat{P}(x_n \mid c_i) = 0$ during test

   – For a remedy, conditional probabilities are estimated with

$$\hat{P}(X_j = a_{jk} \mid C = c_i) = \frac{n_c + mp}{n + m}$$

n_c: number of training examples for which $X_j = a_{jk}$ and C $= c_i$

n: number of training examples for which $C = c_i$

p: prior estimate (usually, $p = 1/t$ for t possible values of X_j)

m: weight to prior (number of "virtual" examples, $m \geq 1$)

Continuous-valued Input Attributes

- What to do in the case of Continuous Valued Inputs?
  - Numberless values for an attribute
  - Conditional probability is then modeled with the normal distribution

$$\hat{P}(X_j|C = c_i) = \frac{1}{\sqrt{2\pi}\sigma_{ji}} \exp\left(-\frac{(X_j - \mu_{ji})^2}{2\sigma_{ji}^2}\right)$$

μ_{ji} : mean (avearage) of attribute values X_j of examples for which $C = c_i$

σ_{ji} : standard deviation of attribute values X_j of examples for which $C = c_i$

  - Learning Phase: for $\mathbf{X} = (X_1, \cdots, X_n)$, $C = c_1, \cdots, c_L$

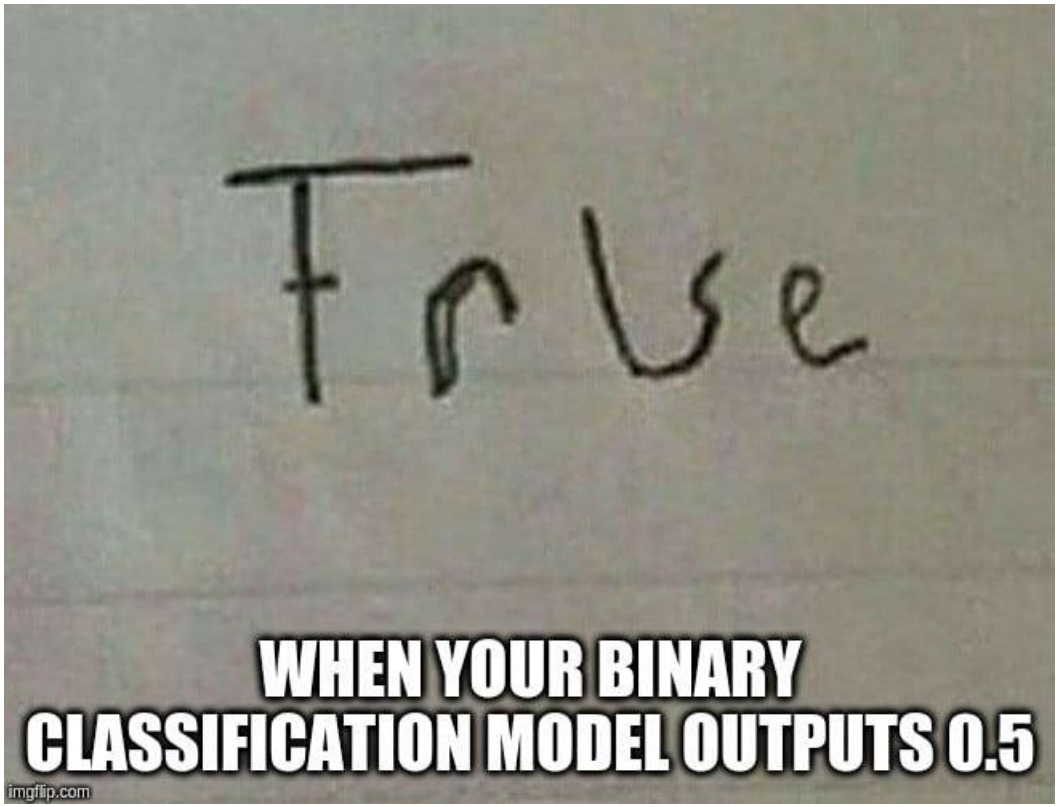
    Output: $n \times L$ normal distributions and $P(C = c_i)$ $i = 1, \cdots, L$
  - Test Phase: for $\mathbf{X}' = (X_1', \cdots, X_n')$
    1. Calculate conditional probabilities with all the normal distributions
    2. Apply the MAP rule to make a decision

Summary

- Naïve Bayes is based on the independence assumption
  - Training is easy and fast; just requiring considering each attribute in each class separately
  - Test is straightforward; just looking up tables or calculating conditional probabilities with normal distributions

- Naïve Bayes is a popular generative classifier model
  1. Performance of naïve Bayes is competitive to most of state-of-the-art classifiers even in presence of violating independence assumption
  2. Suitable for very high-dimensional datasets.
  3. It has many successful applications, e.g., spam mail filtering
  4. A good candidate of a base learner in ensemble learning
  5. Apart from classification, naïve Bayes can do more… ex. handle missing data

Questions?



https://colab.research.google.com/github/jakevdp/PythonDataScienceHandbook/blob/master/notebooks/05.05-Naive-Bayes.ipynb#scrollTo=DUcApV43WTvE