CSE 4553
Machine Learning
Lecture 5: Probabilistic Classifier

Winter 2024

Hasan Mahmud | hasan@iut-dhaka.edu

Contents

- Introduction to probabilistic classifier
- Forms of probabilistic model
- Review of probability
- Naïve Bayes classifier
- Applications

Introduction

- Probabilistic classification means that the model used for classification is a probabilistic model.
- Probabilistic model can give probability of an instance belonging to positive or negative class. Then it is up to us to decide whether the instance is positive or negative based on the probabilities given by the model.

Probabilistic classifier

Input: $S_{\text{train}} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$ training examples

$$y_i \in \{c_1, c_2, \ldots, c_J\}$$

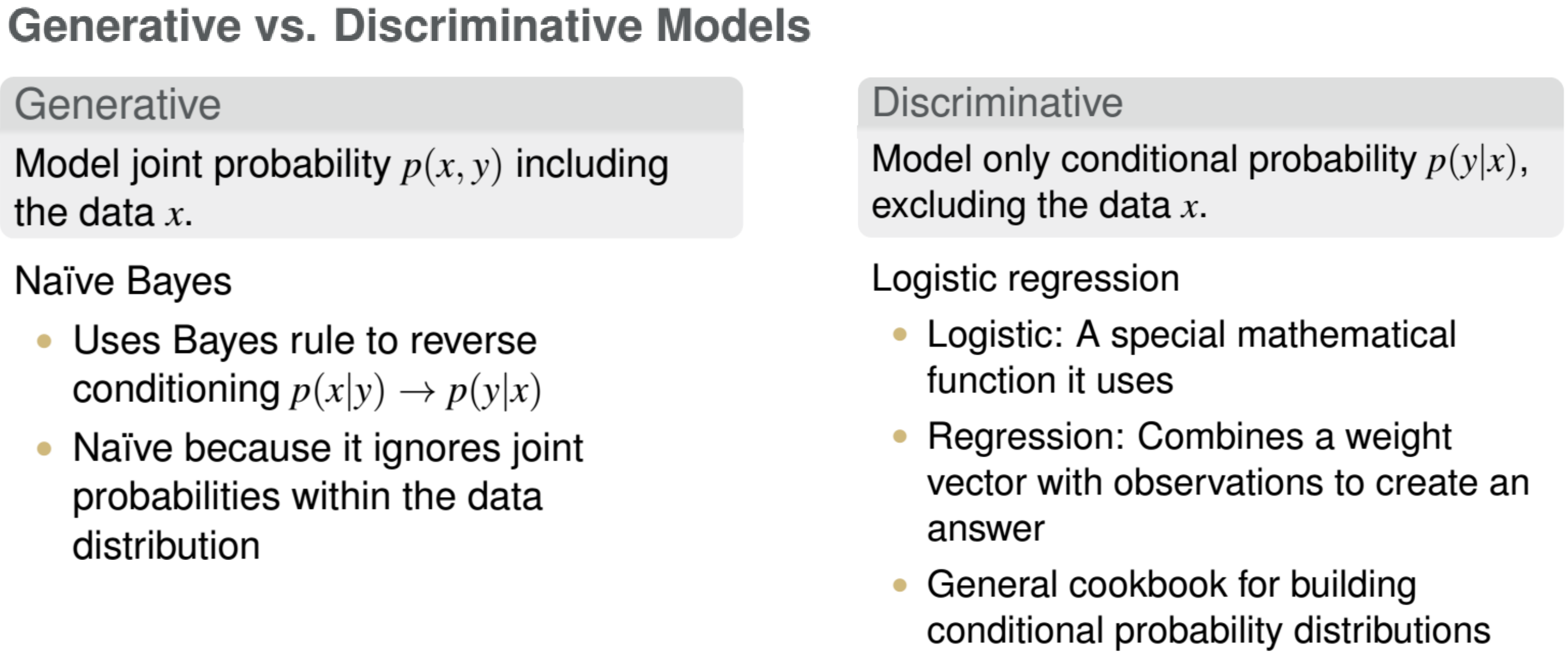
Goal: $h : X \to Y$
For each class c_j, estimate

$$P(y = c_j \mid \boldsymbol{x}, S_{\text{train}})$$

Assign to $\boldsymbol{x}$ the class with the highest probability

$$\hat{y} = h(\boldsymbol{x}) = \arg\max_c P(y = c \mid \boldsymbol{x}, S_{\text{train}})$$

Forms of probabilistic models



Generative vs. Discriminative Models

Generative

Model joint probability $p(x, y)$ including the data x.

Naïve Bayes

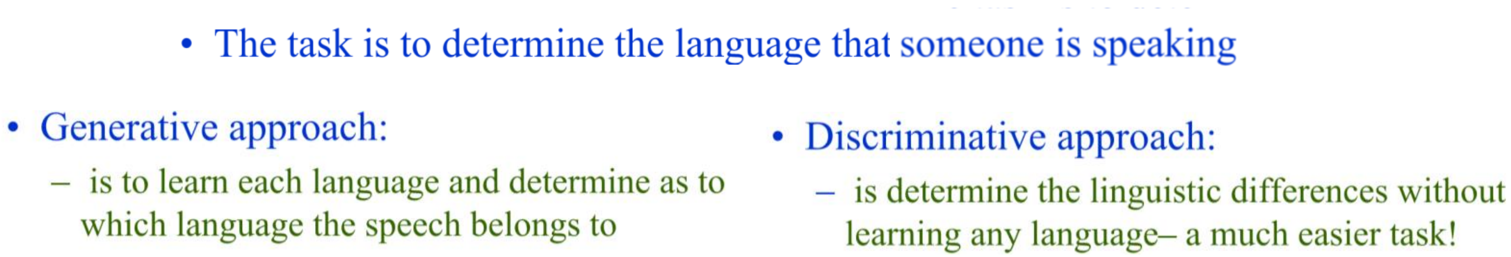
- Uses Bayes rule to reverse conditioning $p(x|y) \rightarrow p(y|x)$
- Naïve because it ignores joint probabilities within the data distribution

Discriminative

Model only conditional probability $p(y|x)$, excluding the data x.

Logistic regression

- Logistic: A special mathematical function it uses
- Regression: Combines a weight vector with observations to create an answer
- General cookbook for building conditional probability distributions



- The task is to determine the language that someone is speaking

- Generative approach:
  - is to learn each language and determine as to which language the speech belongs to
- Discriminative approach:
  - is determine the linguistic differences without learning any language– a much easier task!

Forms of probabilistic models

Two approaches to classification:

- Discriminative classifiers estimate parameters of decision boundary/class separator directly from labeled examples
  - learn $p(y|\mathbf{x})$ directly (logistic regression models)
  - learn mappings from inputs to classes (least-squares, neural nets)

- Generative approach: model the distribution of inputs characteristic of the class (Bayes classifier)
  - Build a model of $p(\mathbf{x}|y)$
  - Apply Bayes Rule

Forms of probabilistic model

- Based on the taxonomy, we can see the essence of different supervised learning models (classifiers) more clearly.

	Probabilistic	**Non-Probabilistic**
Discriminative	• Logistic Regression • Probabilistic neural nets • ……..	• K-nn • Linear classifier • SVM • Neural networks • ……
Generative	• Naïve Bayes • Model-based (e.g., GMM) • ……	**N.A. (?)**

Probability Basics

- Prior, conditional and joint probability for random variables
  - Prior probability: $P(x)$
  - Conditional probability: $P(x_1 | x_2), P(x_2 | x_1)$
  - Joint probability: $\mathbf{x} = (x_1, x_2), P(\mathbf{x}) = P(x_1, x_2)$
  - Relationship: $P(x_1, x_2) = P(x_2 | x_1)P(x_1) = P(x_1 | x_2)P(x_2)$
  - Independence:

$$P(x_2 | x_1) = P(x_2), P(x_1 | x_2) = P(x_1), P(x_1, x_2) = P(x_1)P(x_2)$$

- Bayesian Rule

$$P(c | \mathbf{x}) = \frac{P(\mathbf{x} | c)P(c)}{P(\mathbf{x})}$$

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

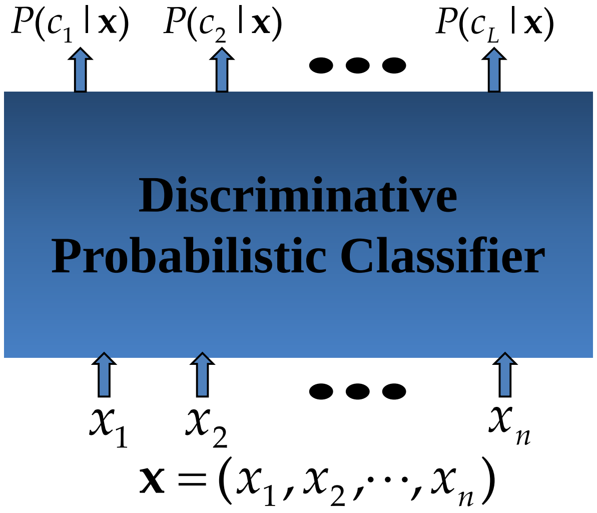
Discriminative

Generative

Probabilistic Classification Principle

- Establishing a probabilistic model for classification
  - **Discriminative model**

$$P(c\,|\,\mathbf{x}) \quad c = c_1, \cdots, c_L, \mathbf{x} = (x_1, \cdots, x_n)$$

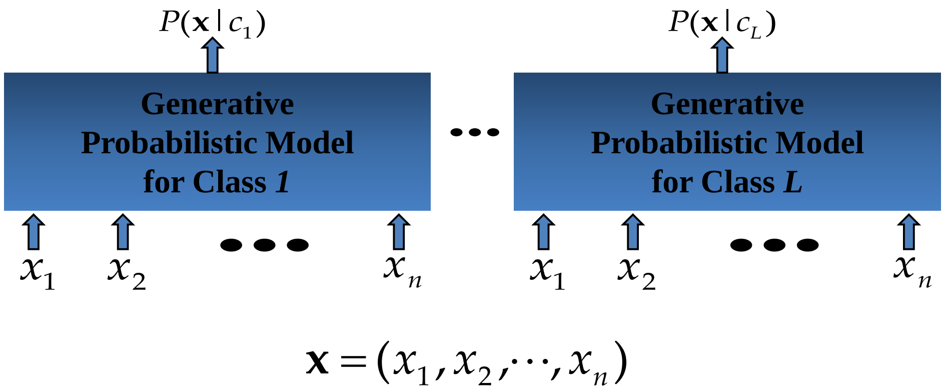


- To train a discriminative classifier (regardless its probabilistic or non-probabilistic nature), all training examples of different classes must be jointly used to build up a single discriminative classifier.
- Output *L* probabilities for *L* class labels in a probabilistic classifier while a single label is achieved by a non-probabilistic discriminative classifier .

Probabilistic Classification Principle

- Establishing a probabilistic model for classification (cont.)
  - **Generative model (must be probabilistic)**

$$\boxed{P(\mathbf{x}\,|\,c)\quad c=c_1,\cdots,c_L,\,\mathbf{x}=(x_1,\cdots,x_n)}$$


$P(\mathbf{x}\,|\,c_1)$
$P(\mathbf{x}\,|\,c_L)$
Generative Probabilistic Model for Class *1*
Generative Probabilistic Model for Class *L*
x_1
x_2
x_n
x_1
x_2
x_n
$\mathbf{x}=(x_1,x_2,\cdots,x_n)$

- *L* probabilistic models have to be trained independently
- Each is trained on only the examples of the same label
- Output *L* probabilities for a given input with *L* models
- "Generative" means that such a model can produce data subject to the distribution via sampling.

Probabilistic Classification Principle

- **Maximum A Posterior (MAP)** classification rule
  - For an input $\boldsymbol{x}$, find the largest one from L probabilities output by a discriminative probabilistic classifier $P(c_1 \mid \mathbf{x}), \ldots, P(c_L \mid \mathbf{x})$.
  - Assign $\boldsymbol{x}$ to label c^* if $P(c^* \mid \mathbf{x})$ is the largest.
- Generative classification with the MAP rule
  - Apply Bayesian rule to convert them into posterior probabilities

$$P(c_i \mid \mathbf{x}) = \frac{P(\mathbf{x} \mid c_i)P(c_i)}{P(\mathbf{x})} \propto P(\mathbf{x} \mid c_i)P(c_i)$$

$$\text{for } i = 1, 2, \cdots, L$$

Common factor for all L probabilities

  - Then apply the MAP rule to assign a label

Naïve Bayes

- Bayes classification

$$P(c\,|\,\mathbf{x}) \propto P(\mathbf{x}\,|\,c)P(c) = P(x_1,\cdots,x_n\,|\,c)P(c) \;\; \text{for}\; c = c_1,\ldots,c_L.$$

Difficulty: learning the joint probability $P(x_1,\cdots,x_n\,|\,c)$ is often infeasible!

- Naïve Bayes classification
  - $P(x_1,x_2,\cdots,x_n\,|\,c) = \underline{P(x_1\,|\,x_2,\cdots,x_n,c)}P(x_2,\cdots,x_n\,|\,c)$ ndependent!

    $= \underline{P(x_1\,|\,c)}P(x_2,\cdots,x_n\,|\,c)$

    $= P(x_1\,|\,c)P(x_2\,|\,c)\cdots P(x_n\,|\,c)$

    Applying the independence assumption

    $\mathbf{x}' = (a_1,a_2,\cdots,a_n)$
  - $\underbrace{[P(a_1\,|\,c^*)\cdots P(a_n\,|\,c^*)]}_{\text{estimate of } P(a_1,\cdots,a_n\,|\,c^*)}P(c^*) > \underbrace{[P(a_1\,|\,c)\cdots P(a_n\,|\,c)]}_{\text{esitmate of } P(a_1,\cdots,a_n\,|\,c)}P(c), \;\; c \neq c^*, c = c_1,\cdots,c_L$ c^* if

Naïve Bayes

- Algorithm: Discrete-Valued Features
  - Learning Phase: Given a training set **S** of F features and L classes,

```
For each target value of $c_i$ $(c_i = c_1, \cdots, c_L)$
    $\hat{P}(c_i) \leftarrow$ estimate $P(c_i)$ with examples in S;
    For every feature value $x_{jk}$ of each feature $x_j$ $(j = 1, \cdots, F; k = 1, \cdots, N_j)$
        $\hat{P}(x_j = x_{jk} | c_i) \leftarrow$ estimate $P(x_{jk} | c_i)$ with examples in S;
```

    Output: $F * L$ conditional probabilistic (generative) models

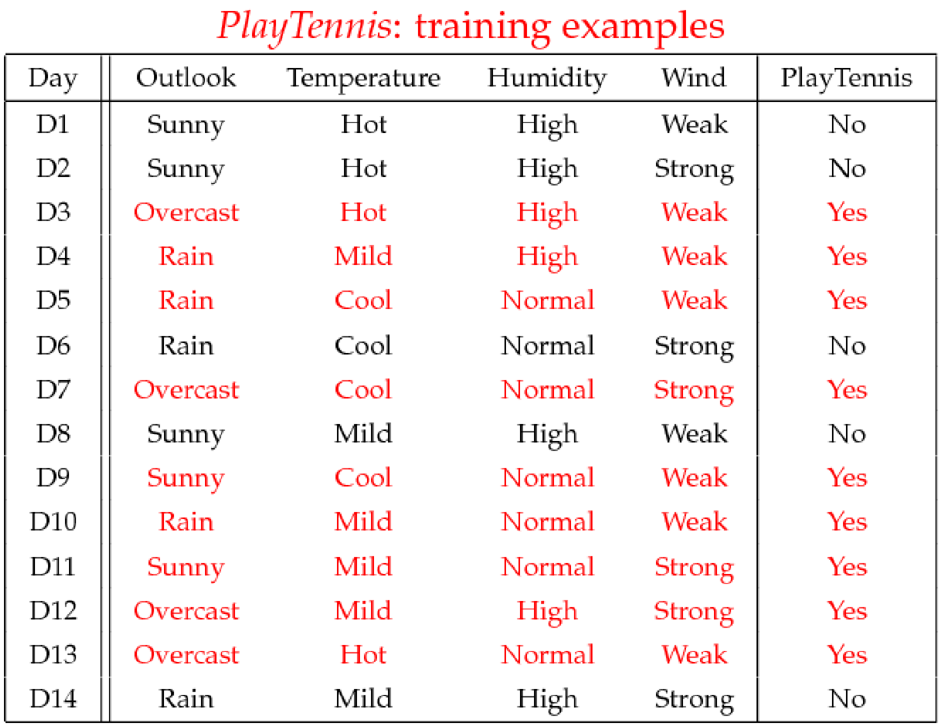
  - Test Phase: Given an unknown instance $\mathbf{x}' = (a'_1, \cdots, a'_n)$

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

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

Example

- Example: Play Tennis

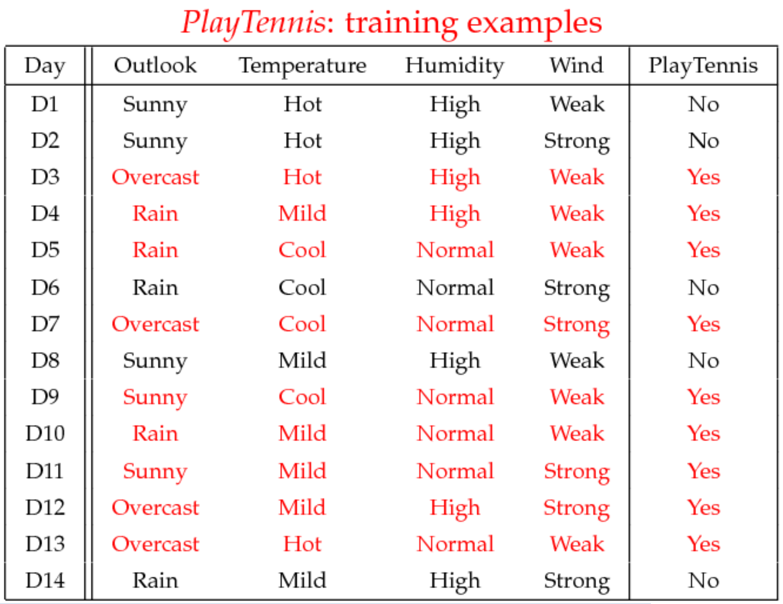


PlayTennis: training examples

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

Example

- Learning Phase: Learning the class distributions
- Goal: Learn the joint probability distribution



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

Likelihood

P(outlook|Yes) P(outlook|No)

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

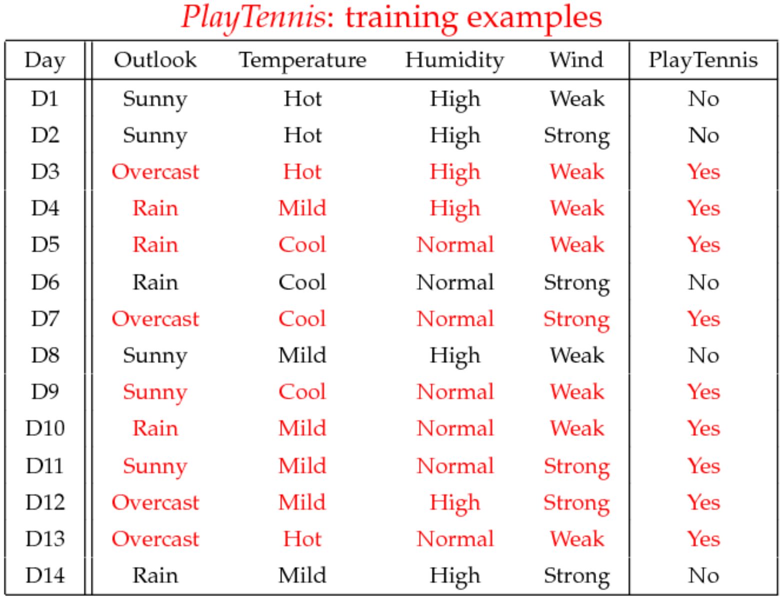
Wind	Play=*Yes*	Play=*No*
Strong	3/9	3/5
Weak	6/9	2/5

Learning Conditional Probabilities

$P(\text{Play}=Yes) = 9/14$ $P(\text{Play}=No) = 5/14$

Prior

Example



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

- Test Phase
  - Given a new instance, predict its label

    $\mathbf{x}'$=(Outlook=*Sunny*, Temperature=*Cool*, Humidity=*High*, Wind=*Strong*)

  - Look up tables achieved in the learning phrase

P(Outlook=*Sunny*|Play=*Yes*) = 2/9	P(Outlook=*Sunny*|Play=*No*) = 3/5
P(Temperature=*Cool*|Play=*Yes*) = 3/9	P(Temperature=*Cool*|Play==*No*) = 1/5
P(Huminity=*High*|Play=*Yes*) = 3/9	P(Huminity=*High*|Play=*No*) = 4/5
P(Wind=*Strong*|Play=*Yes*) = 3/9	P(Wind=*Strong*|Play=*No*) = 3/5
P(Play=*Yes*) = 9/14	P(Play=*No*) = 5/14

  - Decision making with the MAP rule

$$P(Yes|\mathbf{x}') \approx [P(Sunny|Yes)P(Cool|Yes)P(High|Yes)P(Strong|Yes)]\, P(\text{Play}=Yes) = 0.0053$$

$$P(No|\mathbf{x}') \approx [P(Sunny|\text{No})\, P(Cool|\text{No})P(High|No)P(Strong|No)]P(\text{Play}=No) = 0.0206$$

Given the fact $P(Yes|\mathbf{x}') < P(No|\mathbf{x}')$, we label $\mathbf{x}'$ to be "*No*".

Naïve Bayes Classifier

- It is a classification technique based on Bayes' Theorem with an assumption of independence among predictors.
- In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.
- For example, a fruit may be considered to be an apple if it is red, round, and about 3 inches in diameter. Even if these features depend on each other or upon the existence of the other features, all of these properties independently contribute to the probability that this fruit is an apple and that is why it is known as 'Naive'.
- Naive Bayes model is easy to build and particularly useful for very large data sets. Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods.

Classification with Bayes

- Given an instance $\mathbf{x} = (x_1, \ldots, x_p)$, and any class value $c \in \{ 1, \ldots, k \}$, Bayes theorem gives us

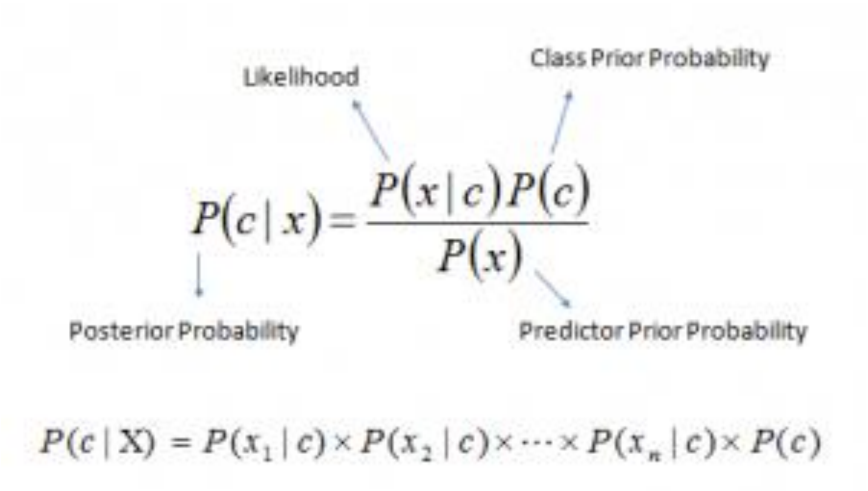
$$P(Y = c \mid X = \mathbf{x}) = \frac{P(X = \mathbf{x} \mid Y = c)P(Y = c)}{\sum_{c'=1}^{k} P(X = \mathbf{x} \mid Y = c')P(Y = c')}$$

- The basic version of naive Bayes then predicts class c with maximum posterior probability (MAP):

$$\hat{c}(\mathbf{x}) = \arg\max_{c} P(Y = c \mid X = \mathbf{x})$$

- Probabilistic predictions are obtained directly from $P(Y = c \mid X = \mathbf{x})$

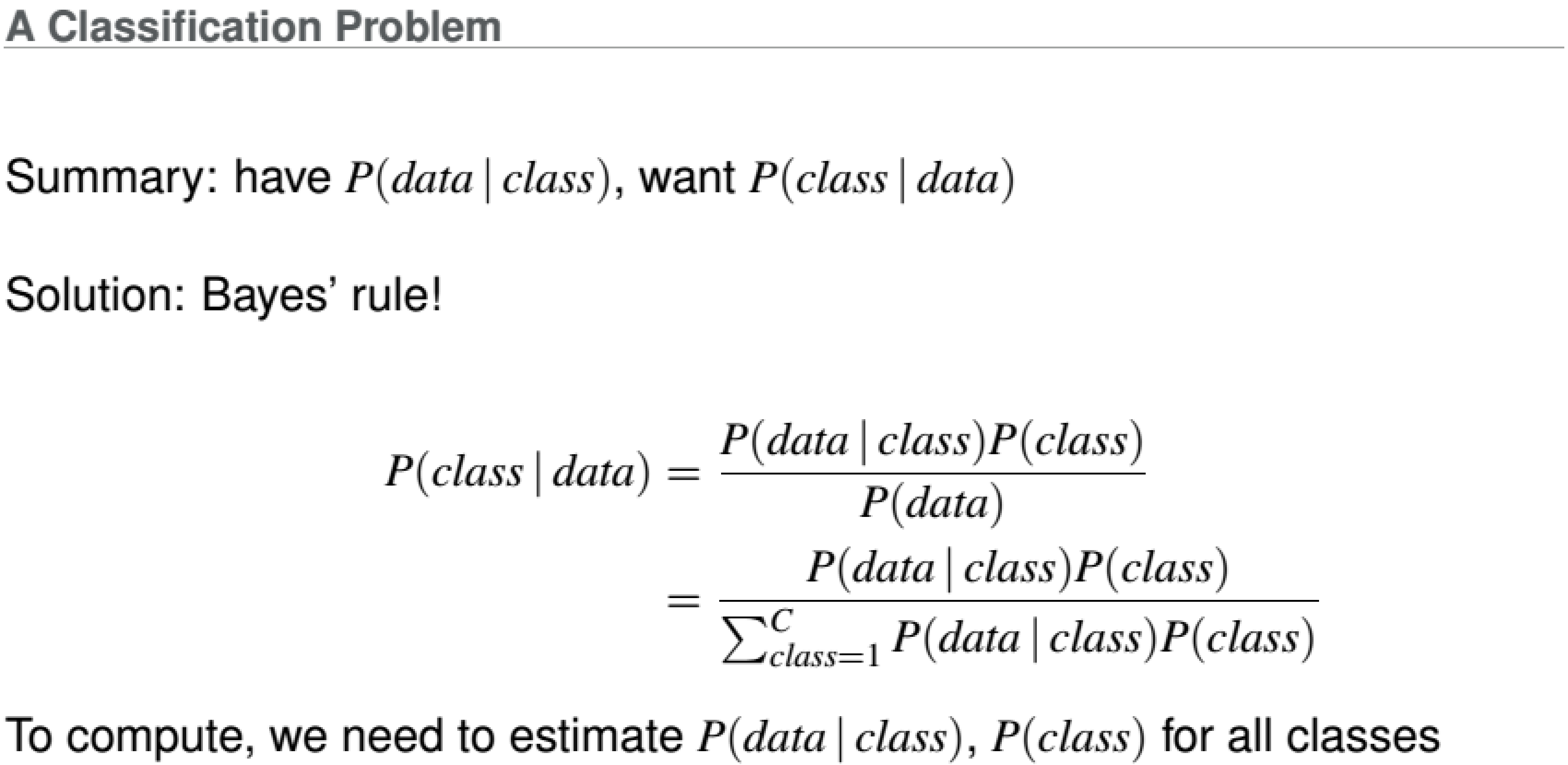
- Bayes theorem provides a way of calculating posterior probability P(c|x) from P(c), P(x) and P(x|c). Look at the equation below:



$$P(c|x) = \frac{P(x|c)P(c)}{P(x)}$$

$$P(c|\mathrm{X}) = P(x_1|c) \times P(x_2|c) \times \cdots \times P(x_n|c) \times P(c)$$

- *P*(*c|x*) is the posterior probability of *class* (c, *target*) given *predictor* (x, *attributes*).
- *P*(*c*) is the prior probability of *class*.
- *P*(*x|c*) is the likelihood which is the probability of *predictor* given *class*.
- *P*(*x*) is the prior probability of *predictor*.



A Classification Problem

Summary: have $P(data \mid class)$, want $P(class \mid data)$

Solution: Bayes' rule!

$$P(class \mid data) = \frac{P(data \mid class)P(class)}{P(data)}$$
$$= \frac{P(data \mid class)P(class)}{\sum_{class=1}^{C} P(data \mid class)P(class)}$$

To compute, we need to estimate $P(data \mid class)$, $P(class)$ for all classes

- The classification is conducted by deriving the maximum posterior which is the maximal P(Ci|**X**) with the above assumption applying to Bayes theorem. This assumption greatly reduces the computational cost by only counting the class distribution. Even though the assumption is not valid in most cases since the attributes are dependent, surprisingly Naive Bayes has able to perform impressively.

$$P(\mathbf{X} \mid C_i) = \prod_{k=1}^{n} P(x_k \mid C_i) = P(x_1 \mid C_i) \times P(x_2 \mid C_i) \times \ldots \times P(x_n \mid C_i)$$

- Application
- Advantages and disadvantages