Probabilistic and Generative Classification Models

Outline

- Background and Probability Basics
- Probabilistic Classification Principle
  - Probabilistic discriminative models
  - Generative models and their application to classification
  - MAP and converting generative into discriminative
- Naïve Bayes – a generative model
  - Principle and Algorithms (discrete vs. continuous)
  - Example: Play Tennis
- Zero Conditional Probability

Background

- There are three methodologies:

a) Model a classification rule directly

Examples: trees, neural nets, ...

b) Model the probability of class memberships given input data

Examples: logistic regression, probabilistic neural nets (softmax),...

c) Make a probabilistic model of data within each class

Examples: naive Bayes, model-based, ...

- Important ML taxonomy for learning models

probabilistic models vs non-probabilistic models

discriminative models vs generative models

Background

- 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 • Classification Trees • 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: $x=(x_1,x_2), P(x)=P(x_1,x_2)$
  - Relationship: $P(x_1,x_2)=P(x_2|x_1)P(x_1)=P(x_1|x_2)P(x_2)$
  - Independence: $P(x_2|x_1)=P(x_2), P(x_1|x_2)=P(x_1), P(x_1,x_2)=P(x_1)P(x_2)$
- Bayes Rule

$$P(c|x)=\frac{P(x|c)P(c)}{P(x)} \qquad Posterior=\frac{Likelihood\times Prior}{Evidence}$$

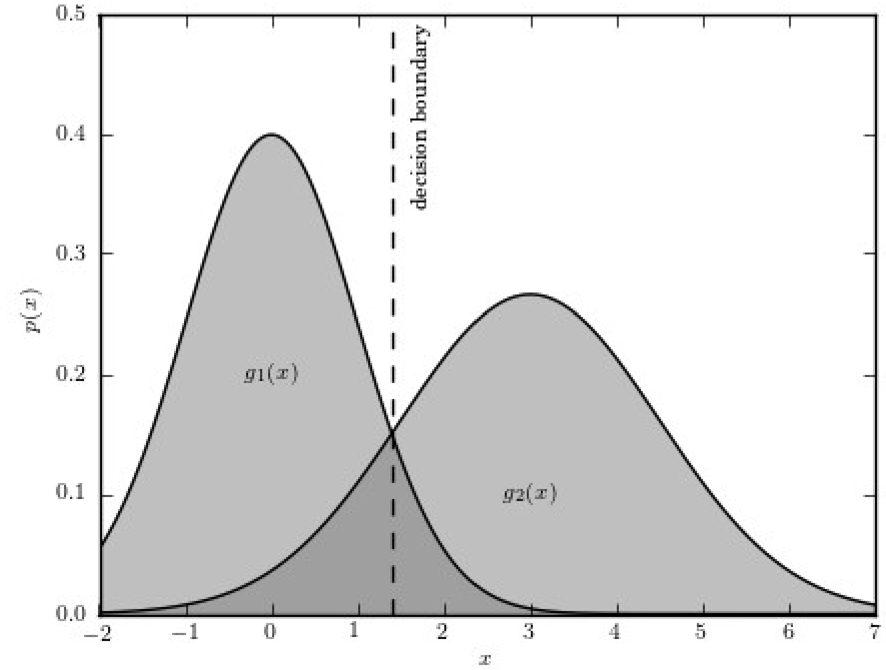
Probabilistic Classification

- Toy example
  - Two classes, two features problems
  - Somebody gives me the probability of each class for every point in space
  - What can I do?

Probabilistic Classification

- Toy example
  - Two classes, two features problems
  - Somebody gives me the probability of each class for every point in space
  - What can I do?

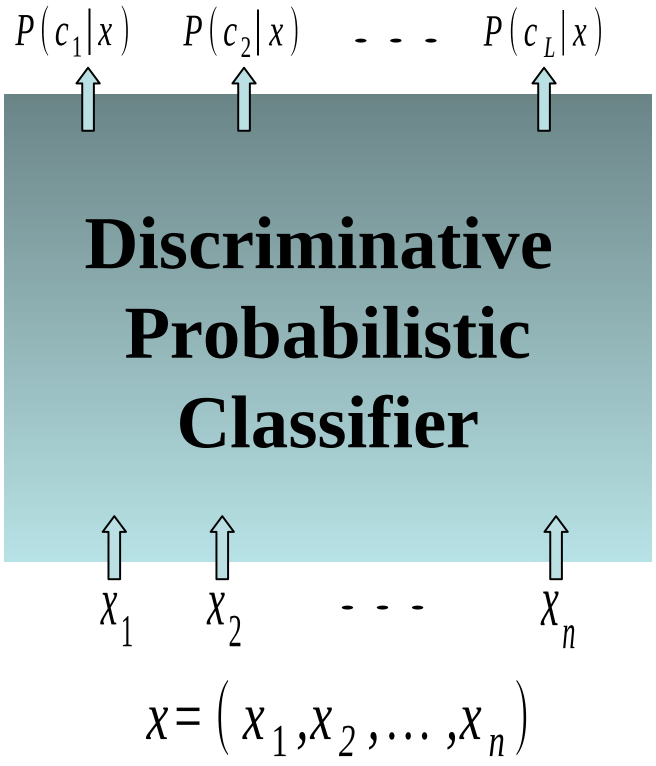
- Assign each class with the max prob.
- “Bayes error” or “Bayes level”



Probabilistic Classification Principle

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

$$P(c|x) \quad c = c_1, \ldots, c_L, \; x = (x_1, \ldots, x_n)$$



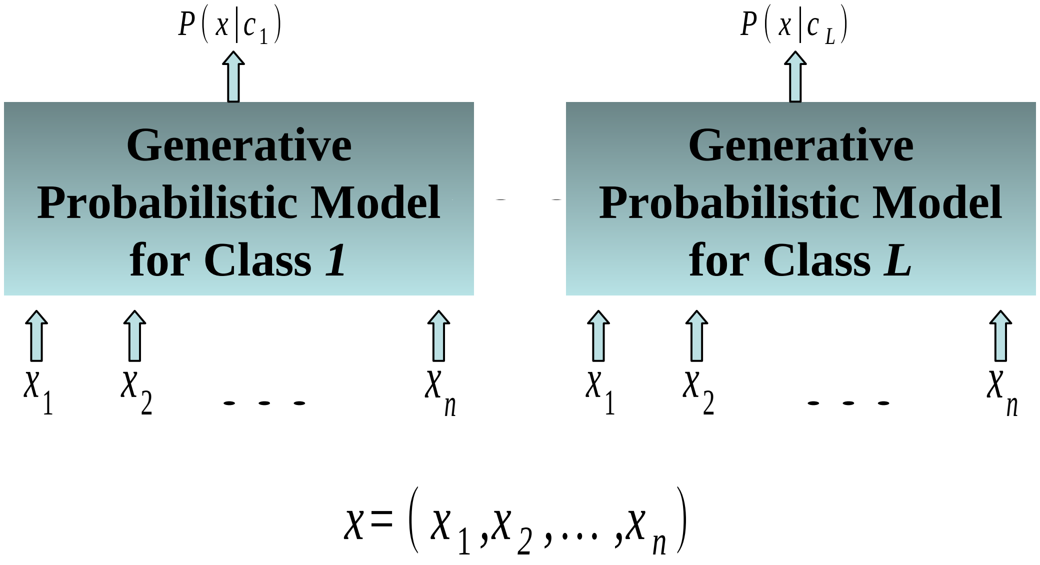
$P(c_1|x)$
$P(c_2|x)$
- - -
$P(c_L|x)$
Discriminative Probabilistic Classifier
x_1
x_2
- - -
x_n
$x = (x_1, x_2, \ldots, x_n)$

- 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**

$$P(x|c) \quad c = c_1, \dots, c_L, \; x = (x_1, \dots, x_n)$$



$P(x|c_1)$
$P(x|c_L)$
Generative Probabilistic Model for Class *1*
Generative Probabilistic Model for Class *L*
x_1
x_2
- - -
x_n
x_1
x_2
- - -
x_n
$x = (x_1, x_2, \dots, x_n)$

- *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

- **M**aximum **A P**osteriori (**MAP**) classification rule
  - For an input $\boldsymbol{x}$, find the largest one from L probabilities output by a discriminative probabilistic classifier $P(c_1|x)$, …, $P(c_L|x)$.
  - Assign $\boldsymbol{x}$ to label c_l if $P(c_l|x)$ is the largest.

Probabilistic Classification Principle

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

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

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

  - Then apply the MAP rule to assign a label

Probabilistic Classification Principle

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

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

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

Common factor for all *L* probabilities

  - Then apply the MAP rule to assign a label

Example of Bayes rule

Does patient have cancer or not?

A patient takes a lab test and the result comes back positive. The test returns a correct positive result (+) in only 98% of the cases in which the disease is actually present, and a correct negative result (-) in only 97% of the cases in which the disease is not present

Furthermore, 0.008 of the entire population have this cancer

Suppose a positive result (+) is returned...

$P(cancer) = 0.008$ $\qquad$ $P(\neg cancer) = 0.992$

$P(+|cancer) = 0.98$ $\qquad$ $P(-|cancer) = 0.02$

$P(+|\neg cancer) = 0.03$ $\qquad$ $P(-|\neg cancer) = 0.97$

$$P(+|cancer) \cdot P(cancer) = 0.98 \cdot 0.008 = 0.0078$$

$$P(+|\neg cancer) \cdot P(\neg cancer) = 0.03 \cdot 0.992 = 0.0298$$

$$h_{MAP} = \neg cancer$$

Normalization

$$\frac{0.0078}{0.0078 + 0.0298} = 0.20745 \qquad \frac{0.0298}{0.0078 + 0.0298} = 0.79255$$

The result of Bayesian inference depends strongly on the prior probabilities, which must be available in order to apply the method

Normalization

$$\frac{0.0078}{0.0078 + 0.0298} = 0.20745 \qquad \frac{0.0298}{0.0078 + 0.0298} = 0.79255$$

The result of ALL MACHINE LEARNING METHODS depends strongly on the prior probabilities, which must be available in order to apply the method

Bayes learning

- Bayes classification

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

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

Naïve Bayes

- Naïve Bayes classification
  - Assume all input features are class conditionally independent!

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

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

$$=P(x_1|c)P(x_2|c)\ldots P(x_n|c)$$

  - Apply the MAP classification rule: assign $x'=(a_1,a_2,\ldots,a_n)$ to c if

$$[P(a_1|c)\ldots P(a_n|c)]P(c)>[P(a_1|c_i)\cdot P(a_n|c_i)]P(c_i),\quad c\neq c_i, \mathrm{c}=\mathrm{c}_1,\ldots,c_L$$

Naïve Bayes

- Naïve Bayes classification
  - Assume all input features are class conditionally independent!

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

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

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

Applying the independence assumption

  - Appl AP classification rule: assign $x'=(a_1,a_2,\dots,a_n)$

$$[P(a_1|c)\dots P(a_n|c)]P(c) > [P(a_1|c_i)\cdot P(a_n|c_i)]P(c_i), \quad c \neq c_i, c=c_1,\dots,c_L$$

Naïve Bayes

- Naïve Bayes classification
  - Assume all input features are class conditionally independent!

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

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

$$=P(x_1|c)P(x_2|c)\ldots P(x_n|c)$$

  - Apply the MAP classification rule: assign $x'=(a_1,a_2,\ldots,a_n)$ to c if

$$[P(a_1|c)\ldots P(a_n|c)]P(c)>[P(a_1|c_i)\cdot P(a_n|c_i)]P(c_i),\quad c\neq c_i, \mathrm{c}=\mathrm{c}_1,\ldots,c_L$$

Naïve Bayes

- Naïve Bayes classification
  - Assume all input features are class conditionally independent!

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

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

$$=P(x_1|c)P(x_2|c)\dots P(x_n|c)$$

  - Apply the MAP classification rule: assign $x'=(a_1,a_2,\dots,a_n)$ to c if

$$\underbrace{[P(a_1|c)\dots P(a_n|c)]}_{\text{estimate of } P(a_1,\dots,a_n|c)}P(c)>\underbrace{[P(a_1|c_i)\cdot P(a_n|c_i)]}_{\text{estimate of } P(a_1,\dots,a_n|c_i)}P(c_i),\quad c\neq c_i,\mathrm{c}=\mathrm{c}_1,\dots,c_L$$

Naïve Bayes

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

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

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

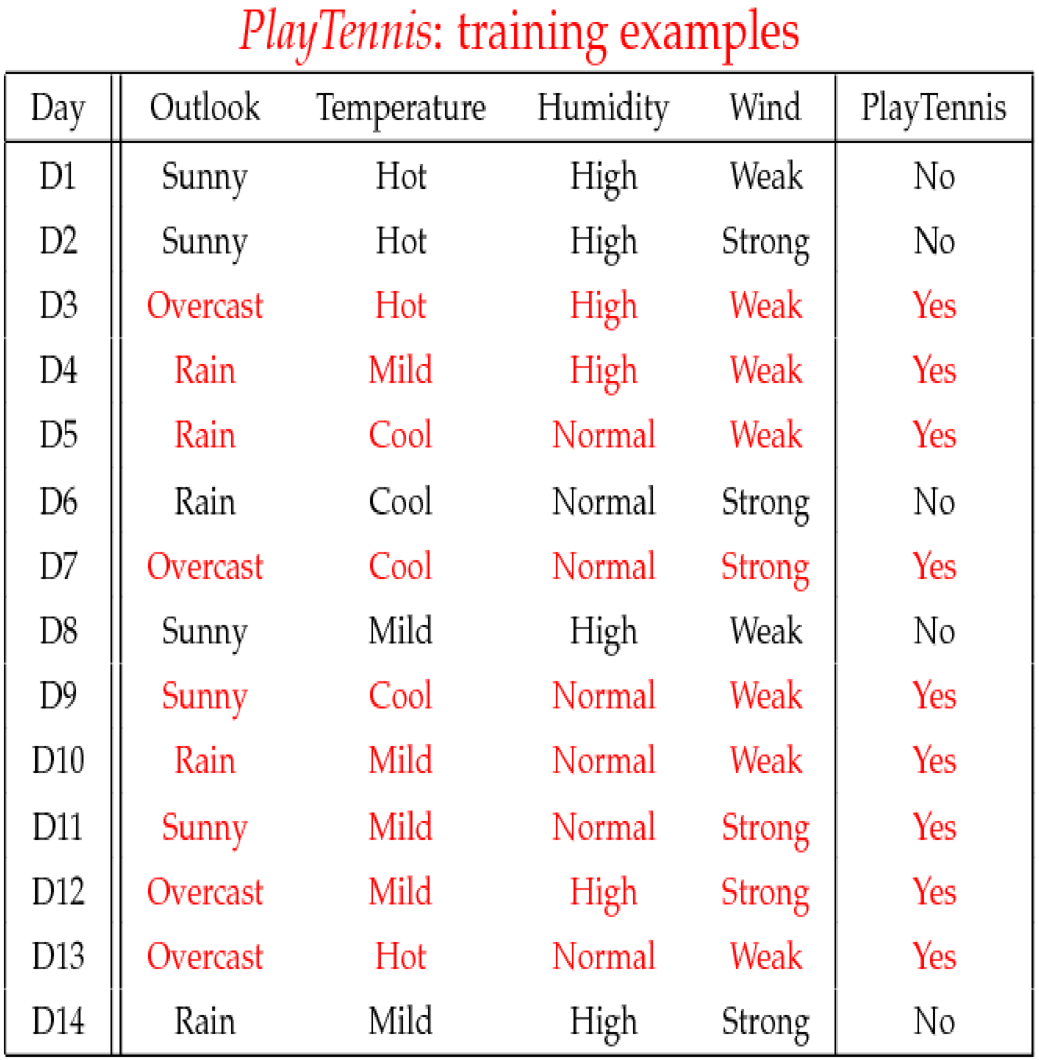
  - Test Phase: Given an unknown instance $x' = \left(a'_1, \dots, a'_n\right)$

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

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

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

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

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

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

Example

- 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

$P(\text{Outlook}=\mathit{Sunny}|\text{Play}=\mathit{Yes}) = 2/9$

$P(\text{Temperature}=\mathit{Cool}|\text{Play}=\mathit{Yes}) = 3/9$

$P(\text{Huminity}=\mathit{High}|\text{Play}=\mathit{Yes}) = 3/9$

$P(\text{Wind}=\mathit{Strong}|\text{Play}=\mathit{Yes}) = 3/9$

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

$P(\text{Outlook}=\mathit{Sunny}|\text{Play}=\mathit{No}) = 3/5$

$P(\text{Temperature}=\mathit{Cool}|\text{Play}==\mathit{No}) = 1/5$

$P(\text{Huminity}=\mathit{High}|\text{Play}=\mathit{No}) = 4/5$

$P(\text{Wind}=\mathit{Strong}|\text{Play}=\mathit{No}) = 3/5$

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

  - Decision making with the MAP rule

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

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

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

Zero conditional probability

- If no example contains the feature value
  - In this circumstance, we face a zero conditional probability problem during test
    $\hat{P}(x_1|c_i)\ldots\hat{P}(a_{jk}|c_i)\ldots\hat{P}(x_n|c_i)=0$
  - For a remedy, class conditional probabilities re-estimated with

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

(m-estimate)

Zero conditional probability

- Example: P(outlook=overcast|no)=0 in the play-tennis dataset
  - Adding m "virtual" examples (m: tunable but low)
    - In this dataset, # of training examples for the "no" class is 5.
    - Assume that we add m**=1** "virtual" example in our m-estimate treatment.
  - The "outlook" feature can takes only 3 values. So $p=1/3$.
  - Re-estimate P(outlook|no) with the m-estimate

Zero conditional probability

- Example: P(outlook=overcast | no)=0 in the play-tennis dataset
  - Adding m "virtual" examples (m: tunable but low)
    - In this dataset, # of training examples for the "no" class is 5.
    - Assume that we add m**=1** "virtual" example in our m-estimate treatment.
  - The "outlook" feature can takes only 3 values. So p=
  - Re e

$$P(\text{overcast}|\text{no}) = \frac{0+1*\left(\frac{1}{3}\right)}{5+1} = \frac{1}{18}$$

$$P(\text{sunny}|\text{no}) = \frac{3+1*\left(\frac{1}{3}\right)}{5+1} = \frac{5}{9} \qquad P(\text{rain}|\text{no}) = \frac{2+1*\left(\frac{1}{3}\right)}{5+1} = \frac{7}{18}$$

Naïve Bayes

- Algorithm: Continuous-valued Features
  - Conditional probability is often modelled with the normal distribution

$$\hat{P}(x_j | c_i) = \frac{1}{\sqrt{2\pi}\sigma_{ji}} e^{-\frac{(x_j - \mu_{ji})^2}{2\sigma_{ji}^2}}$$

  - Learning Phase: Estimate priors,means and variances for each class and feature from data

    Output: F x L normal distributions and $P(C = c_i) \quad i = 1, \ldots, L$

  - Test Phase: Given an unknown instance $X' = (a'_1, \ldots, a'_F)$
    - Instead of looking-up tables, calculate conditional probabilities with all the normal distributions achieved in the learning phrase
    - Apply the MAP rule to assign a label (the same as done for the discrete case)

Naïve Bayes

- Example: Continuous-valued Features
  - Temperature is naturally of continuous value.

    Yes: 25.2, 19.3, 18.5, 21.7, 20.1, 24.3, 22.8, 23.1, 19.8

    No: 27.3, 30.1, 17.4, 29.5, 15.1
  - Estimate mean and variance for each class

$$\mu = \frac{1}{N}\sum_{n=1}^{N} x_n, \quad \sigma^2 = \frac{1}{N}\sum_{n=1}^{N}(x_n - \mu)^2 \qquad \begin{matrix} \mu_{Yes} = 21.64, & \sigma_{Yes} = 2.35 \\ \mu_{No} = 23.88, & \sigma_{No} = 7.09 \end{matrix}$$

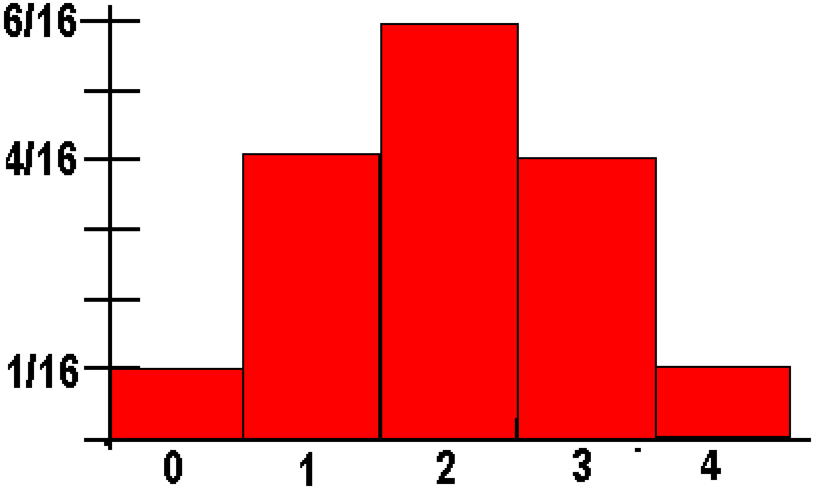
  - **Learning Phase**: output two Gaussian models for P(temp|C)

$$\hat{P}(x|Yes) = \frac{1}{2.35\sqrt{2\pi}}\exp\left(-\frac{(x-21.64)^2}{2\times 2.35^2}\right)$$

$$\hat{P}(x|No) = \frac{1}{7.09\sqrt{2\pi}}\exp\left(-\frac{(x-23.88)^2}{2\times 7.09^2}\right)$$

Naïve Bayes

- Algorithm: Continuous-valued Features
  - Other solution: continuous variables → discrete values
  - Histogram!



Naïve Bayes

- Numerical problems
  - The chain product of several small numbers loose precision in floating point computing

$$P\left(c\middle|a_1\right)=\left[P\left(a_1\middle|c\right)\ldots P\left(a_n\middle|c\right)\right]P\left(c\right)$$

  - Better use logs:

$$\ln\left(P\left(c\middle|a_1\right)\right)=\ln\left(\left[P\left(a_1\middle|c\right)\ldots P\left(a_n\middle|c\right)\right]P\left(c\right)\right)$$

$$\ln\left(P\left(c\middle|a_1\right)\right)=\ln\left(\left[P\left(a_1\middle|c\right)\right)+\ldots+\ln\left(P\left(a_n\middle|c\right)\right]\right)+\ln\left(P\left(c\right)\right)$$

Summary

- Probabilistic Classification Principle
  - Discriminative vs. Generative models: learning P(c|x) vs. P(x|c)P(c)
  - Generative models for classification: MAP and Bayesian rule
- Naïve Bayes: the conditional independence assumption
  - Training and test are very efficient.
  - Two different data types lead to two different learning algorithms.

Summary

- Naïve Bayes: a popular generative model for classification
  - Performance competitive to many state-of-the-art classifiers even when the conditional independence assumption is invalid.
  - Many successful applications, e.g., spam mail filtering, …