NAIVE BAYES CLASSIFIER

Dr. M. SRIDEVI

BACKGROUND

- There are three methodologies:

  a) Model a classification rule directly

  Examples: k-NN, linear classifier, SVM, neural nets, …

  b) Make a probabilistic model of data within each class

  Examples: naive Bayes, model-based ….

  c) Model the probability of class memberships given input data

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

- 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 • 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: $P(x_1 \mid x_2), P(x_2 \mid x_1)$
  - Joint probability: $\mathbf{x} = (x_1, x_2)$, $P(\mathbf{x}) = P(x_1, x_2)$
  - Relationship: $P(x_1, x_2) = P(x_2 \mid x_1)P(x_1) = P(x_1 \mid x_2)P(x_2)$
  - Independence:

$$P(x_2 \mid x_1) = P(x_2),\ P(x_1 \mid x_2) = P(x_1),\ P(x_1, x_2) = P(x_1)P(x_2)$$

- Bayesian Rule

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

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

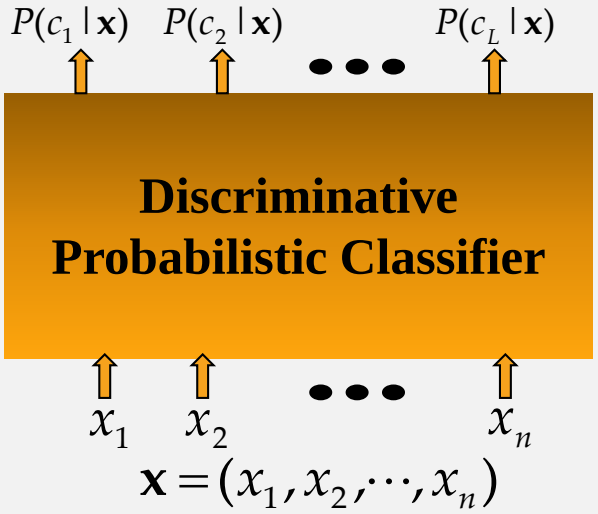
Discriminative

Generative

PROBABILISTIC CLASSIFICATION PRINCIPLE

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

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

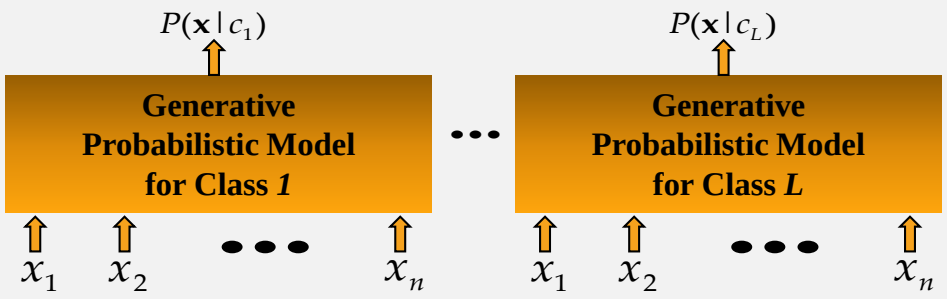


- To train a discriminative classifier (regardless its probabilistic or non-probabilistic nature), all training examples of different classes must be jointly used to build up a single discriminative classifier.
- Output L probabilities for L class labels in a probabilistic classifier while a single label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

PROBABILISTIC CLASSIFICATION PRINCIPLE

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

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



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

- 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

- **M**aximum **A P**osterior (**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

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

Common factor for all L probabilities

  - Then apply the MAP rule to assign a label

GENERAL

The Naive Bayes algorithm is called "**naive**" because it makes the assumption that the occurrence of a certain feature is independent of the occurrence of other features.

Bayes -
It refers to the statistician and philosopher Thomas Bayes and the theorem named after him, Bayes' theorem, which is the base for the Naive Bayes Algorithm

Bayes Theorem -
Bayes' Theorem is stated as Probability of the event B given A is equal to the probability of the event A given B multiplied by the probability of A upon probability of B

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)}$$

- $P(A|B)$: Probability (conditional probability) of occurrence of event A given the event B is true
- $P(A)$ and $P(B)$: Probabilities of the occurrence of event A and B respectively
- $P(B|A)$: Probability of the occurrence of event B given the event A is true

Derivation of Bayes' Theorem

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

$$P(B|A) = \frac{P(B \cap A)}{P(A)}$$

$$P(B|A).P(A) = P(A|B).P(B) \Longrightarrow P(A|B) = \frac{P(B|A).P(A)}{P(B)}$$

Bayesian method of probability

- A is called the proposition and is called the evidence
- P(A) is called the prior probability of proposition and P(B) is called the prior probability of evidence
- P(A|B) is called the posterior
- P(B|A) is the likelihood

$$\text{Posterior} = \frac{\text{(Likelihood).(Proposition prior probability)}}{\text{Evidence prior probability}}$$

SIMPLE EXAMPLE

52 Cards

$$P(\text{Queen}) = \frac{4}{52} = \frac{1}{13}$$

$$P(\text{Queen}\mid\text{Face}) = \frac{P(\text{Face}\mid\text{Queen})}{P(\text{Face})}.P(\text{Queen})$$

$$P(\text{Queen}\mid\text{Face}) = \frac{P(\text{Face}\mid\text{Queen})}{P(\text{Face})}.P(\text{Queen})$$

$$P(\text{Face}\mid\text{Queen}) = 1 \quad P(\text{Queen}) = \frac{1}{13} \quad P(\text{Face}) = \frac{3}{13}$$

$$P(\text{Queen}\mid\text{Face}) = \frac{1}{13} \cdot \frac{13}{3} = \frac{1}{3}$$

Bayes' Theorem for Naive Bayes Algorithm

In a machine learning classification problem, there are multiple features and classes, say, C_1, C_2, ..., C_k . The main aim in the Naive Bayes algorithm is to calculate the conditional probability of an object with a feature vector x_1, x_2,....., x_n belongs to a particular class C_i,

$$P(C_i|x_1, x_2, \ldots, x_n) = \frac{P(x_1, x_2, \ldots, x_n|C_i).P(C_i)}{P(x_1, x_2, \ldots, x_n)} \text{ for } 1 \leq i \leq k$$

NAÏVE BAYES

- Bayes classification

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

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

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

Applying the independence assumption

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

$\mathbf{x}' = (a_1, a_2, \cdots, a_n)$

- $[P(a_1 \mid c^*) \cdots P(a_n \mid c^*)]P(c^*) > [P(a_1 \mid c) \cdots P(a_n \mid c)]P(c), \quad c \neq c^*, c = c_1, \cdots, c_L$ c^* if

estimate of $P(a_1, \cdots, a_n \mid c^*)$

esitmate of $P(a_1, \cdots, a_n \mid c)$

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 $(j = 1, \cdots, F; k = 1, \cdots, N_j)$

$\hat{P}(x_j = x_{jk} \mid c_i) \leftarrow$ estimate $P(x_{jk} \mid c_i)$ 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 $\mathbf{X}'$ if

$$[\hat{P}(a_1' \mid c^*) \cdots \hat{P}(a_n' \mid c^*)]\hat{P}(c^*) > [\hat{P}(a_1' \mid c_i) \cdots \hat{P}(a_n' \mid c_i)]\hat{P}(c_i), \quad c_i \neq c^*, c_i = c_1, \cdots, c_L$$

IN GENERAL - MACHINE LEARNING APPROACH

- Like human learning from past experiences.
- A computer does not have "experiences".
- A computer system learns from data, which represent some "past experiences" of an application domain.
- Our focus: learn a target function that can be used to predict the values of a discrete class attribute, e.g., approve or not-approved, and high-risk or low risk.
- The task is commonly called: Supervised learning, classification, or inductive learning.

THE DATA AND THE GOAL

- Data: A set of data records (also called examples, instances or cases) described by
  - k attributes: $A_1, A_2, \ldots A_k$.
  - a class: Each example is labelled with a pre-defined class.
- Goal: To learn a classification model from the data that can be used to predict the classes of new (future, or test) cases/instances.

EXAMPLE 1

- 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 1 (TRAINING)

- 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

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

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

EXAMPLE I (TESTING)

- 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(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

  - Decision making with the MAP rule

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

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

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

EXAMPLE 2

- Compute all probabilities required for classification

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

$Pr(C = t) = 1/2$, $Pr(C = f) = 1/2$

$Pr(A=m \mid C=t) = 2/5$ $Pr(A=g \mid C=t) = 2/5$ $Pr(A=h \mid C=t) = 1/5$

$Pr(A=m \mid C=f) = 1/5$ $Pr(A=g \mid C=f) = 2/5$ $Pr(A=h \mid C=n) = 2/5$

$Pr(B=b \mid C=t) = 1/5$ $Pr(B=s \mid C=t) = 2/5$ $Pr(B=q \mid C=t) = 2/5$

$Pr(B=b \mid C=f) = 2/5$ $Pr(B=s \mid C=f) = 1/5$ $Pr(B=q \mid C=f) = 2/5$

Now we have a test example:

$A = m \quad B = q \quad C = ?$

EXAMPLE 2 (CONT …)

- For C = t, we have

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

- For class C = f, we have

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

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

EXAMPLE 3: DATA (LOAN APPLICATION)

Approved or not

ID	Age	Has_Job	Own_House	Credit_Rating	Class
1	young	false	false	fair	**No**
2	young	false	false	good	**No**
3	young	true	false	good	**Yes**
4	young	true	true	fair	**Yes**
5	young	false	false	fair	**No**
6	middle	false	false	fair	**No**
7	middle	false	false	good	**No**
8	middle	true	true	good	**Yes**
9	middle	false	true	excellent	**Yes**
10	middle	false	true	excellent	**Yes**
11	old	false	true	excellent	**Yes**
12	old	false	true	good	**Yes**
13	old	true	false	good	**Yes**
14	old	true	false	excellent	**Yes**
15	old	false	false	fair	**No**

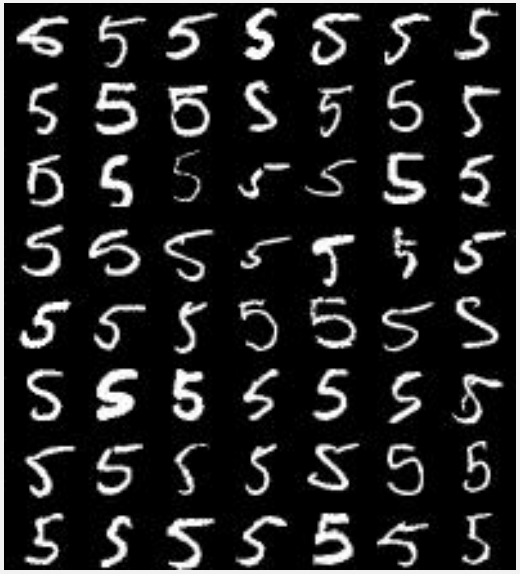
EXAMPLE 3 - LEARNING TASK

- Learn a classification model from the data
- Use the model to classify future loan applications into
  - Yes (approved) and
  - No (not approved)
- What is the class for following case/instance?

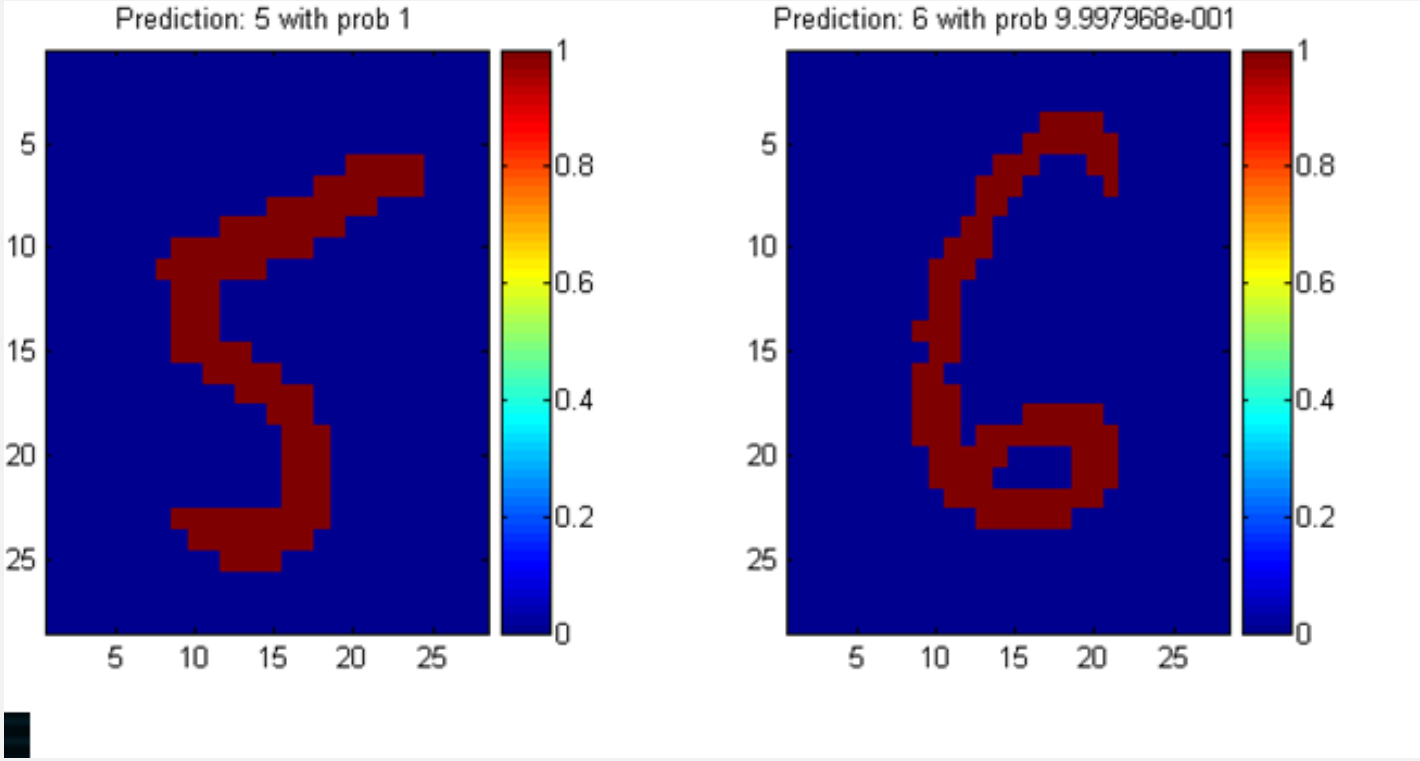
Age	Has_Job	Own_house	Credit-Rating	Class
young	false	false	good	?

EXAMPLE 4
MNIST TRAINING SAMPLE

- Now that we've decided to use a Naïve Bayes classifier, we need to train it with some data: MNIST Training Data



EXAMPLE 4 – TEST RESULT



APPLICATION 1

- An emergency room in a hospital measures 17 variables (e.g., blood pressure, age, etc) of newly admitted patients.
- A decision is needed: whether to put a new patient in an intensive-care unit.
- Due to the high cost of ICU, those patients who may survive less than a month are given higher priority.
- Problem: to predict high-risk patients and discriminate them from low-risk patients.

APPLICATION 2

- A credit card company receives thousands of applications for new cards. Each application contains information about an applicant,
  - age
  - Marital status
  - annual salary
  - outstanding debts
  - credit rating
  - etc.
- Problem: to decide whether an application should approved, or to classify applications into two categories, approved and not approved.

ADVANTAGES AND DISADVANTAGES OF NAÏVE BAYESIAN CLASSIFIER

- Advantages:
  - Easy to implement
  - Very efficient
  - Good results obtained in many applications
- Disadvantages
  - Assumption: class conditional independence, therefore loss of accuracy when the assumption is seriously violated (those highly correlated data sets)

ADDITIONAL ISSUES

- **Numeric attributes**: Naïve Bayesian learning assumes that all attributes are categorical. Numeric attributes need to be discretized.
- **Zero counts**: An particular attribute value never occurs together with a class in the training set. We need smoothing.
- **Missing values**: Ignored

$$\Pr(A_i = a_i \mid C = c_j) = \frac{n_{ij} + \lambda}{n_j + \lambda n_i}$$

Thank you

FOR ADDITIONAL SLIDES GO AHEAD

NAÏVE BAYES

- Algorithm: Continuous-valued Features
  - Numberless values taken by a continuous-valued feature
  - Conditional probability is often modelled with the normal distribution

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

μ_{ji} : mean (avearage) of feature values x_j of examples for which $c = c_i$

σ_{ji} : standard deviation of feature values x_j of examples for which $c = c_i$

  - Learning Phase: for $\mathbf{X} = (X_1, \cdots, X_F)$, $C = c_1, \cdots, c_L$
    Output: $F \times L$ normal distributions and $P(C = c_i)$ $i = 1, \cdots, L$
  - Test Phase: Given an unknown instance $\mathbf{X}' = (a_1', \cdots, a_n')$
    - 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

$$\mu = \frac{1}{N}\sum_{n=1}^{N} x_n, \quad \sigma^2 = \frac{1}{N}\sum_{n=1}^{N}(x_n - \mu)^2$$

$$\mu_{Yes} = 21.64, \ \sigma_{Yes} = 2.35$$
$$\mu_{No} = 23.88, \ \sigma_{No} = 7.09$$

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

$$\hat{P}(x \mid Yes) = \frac{1}{2.35\sqrt{2\pi}} \exp\left(-\frac{(x-21.64)^2}{2\times 2.35^2}\right) = \frac{1}{2.35\sqrt{2\pi}} \exp\left(-\frac{(x-21.64)^2}{11.09}\right)$$

$$\hat{P}(x \mid No) = \frac{1}{7.09\sqrt{2\pi}} \exp\left(-\frac{(x-23.88)^2}{2\times 7.09^2}\right) = \frac{1}{7.09\sqrt{2\pi}} \exp\left(-\frac{(x-23.88)^2}{50.25}\right)$$

ZERO CONDITIONAL PROBABILITY

- If no example contains the feature value
  - In this circumstance, we face a zero conditional probability problem during test

$$\hat{P}(x_1 \mid c_i) \cdots \hat{P}(a_{jk} \mid c_i) \cdots \hat{P}(x_n \mid c_i) = 0 \quad \text{for } x_j = a_{jk},\ \hat{P}(a_{jk} \mid c_i) = 0$$

  - For a remedy, class conditional probabilities re-estimated with

$$\hat{P}(a_{jk} \mid c_i) = \frac{n_c + mp}{n + m}$$ **(m-estimate)**

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, $m \geq 1$)

ZERO CONDITIONAL PROBABILITY

- Example: $P(\text{outlook=overcast}|\text{no})=0$ in the play-tennis dataset
  - Adding m "virtual" examples (m: tunable but up to 1% of #training examples)
    - 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 $P(\text{outlook}|\text{no})$ with the m-estimate

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

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

BAYESIAN CLASSIFICATION

- Probabilistic view: Supervised learning can naturally be studied from a probabilistic point of view.
- Let A_1 through A_k be attributes with discrete values. The class is C.
- Given a test example *d* with observed attribute values a_1 through a_k.
- Classification is basically to compute the following posteriori probability. The prediction is the class c_j such that

$$\Pr(C = c_j \mid A_1 = a_1, \ldots, A_{|A|} = a_{|A|})$$

is maximal

APPLY BAYES' RULE

$$\Pr(C = c_j \mid A_1 = a_1, \ldots, A_{|A|} = a_{|A|})$$

$$= \frac{\Pr(A_1 = a_1, \ldots, A_{|A|} = a_{|A|} \mid C = c_j)\Pr(C = c_j)}{\Pr(A_1 = a_1, \ldots, A_{|A|} = a_{|A|})}$$

$$= \frac{\Pr(A_1 = a_1, \ldots, A_{|A|} = a_{|A|} \mid C = c_j)\Pr(C = c_j)}{\sum_{r=1}^{|C|} \Pr(A_1 = a_1, \ldots, A_{|A|} = a_{|A|} \mid C = c_r)\Pr(C = c_r)}$$

- $\Pr(C=c_j)$ is the class *prior* probability: easy to estimate from the training data.

COMPUTING PROBABILITIES

- The denominator $P(A_1=a_1,...,A_k=a_k)$ is irrelevant for decision making since it is the same for every class.
- We only need $P(A_1=a_1,...,A_k=a_k \mid C=c_i)$, which can be written as

$$Pr(A_1=a_1|A_2=a_2,...,A_k=a_k, C=c_j)* Pr(A_2=a_2,...,A_k=a_k \mid C=c_j)$$

- Recursively, the second factor above can be written in the same way, and so on.
- Now an assumption is needed.

CONDITIONAL INDEPENDENCE ASSUMPTION

- All attributes are conditionally independent given the class $C = c_j$.
- Formally, we assume,

$$\Pr(A_1=a_1 \mid A_2=a_2, \ldots, A_{|A|}=a_{|A|}, C=c_j) = \Pr(A_1=a_1 \mid C=c_j)$$

and so on for A_2 through $A_{|A|}$. I.e.,

$$\Pr(A_1 = a_1, \ldots, A_{|A|} = a_{|A|} \mid C = c_i) = \prod_{i=1}^{|A|} \Pr(A_i = a_i \mid C = c_j)$$

FINAL NAÏVE BAYESIAN CLASSIFIER

$$\Pr(C = c_j \mid A_1 = a_1, \ldots, A_{|A|} = a_{|A|})$$

$$= \frac{\Pr(C = c_j) \prod_{i=1}^{|A|} \Pr(A_i = a_i \mid C = c_j)}{\sum_{r=1}^{|C|} \Pr(C = c_r) \prod_{i=1}^{|A|} \Pr(A_i = a_i \mid C = c_r)}$$

- We are done!
- How do we estimate $P(A_i = a_i \mid C = c_j)$? Easy!.

CLASSIFY A TEST INSTANCE

- If we only need a decision on the most probable class for the test instance, we only need the numerator as its denominator is the same for every class.
- Thus, given a test example, we compute the following to decide the most probable class for the test instance

$$c = \arg\max_{c_j} \Pr(c_j) \prod_{i=1}^{|A|} \Pr(A_i = a_i \mid C = c_j)$$

TEXT CLASSIFICATION/CATEGORIZATION

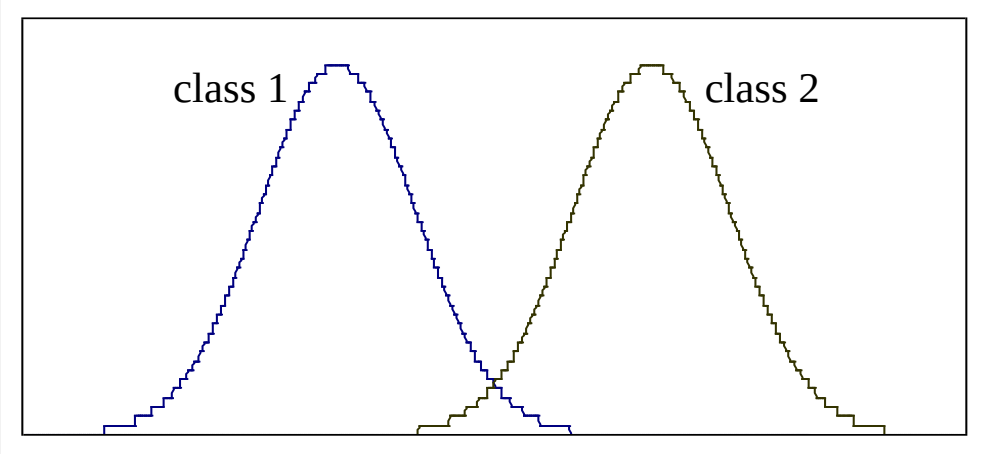
- Due to the rapid growth of online documents in organizations and on the Web, automated document classification has become an important problem.
- Techniques discussed previously can be applied to text classification, but they are not as effective as the next three methods.
- We first study a naïve Bayesian method specifically formulated for texts, which makes use of some text specific features.
- However, the ideas are similar to the preceding method.

MIXTURE MODEL

- A **mixture model** models the data with a number of statistical distributions.
  - Intuitively, each distribution corresponds to a data cluster and the parameters of the distribution provide a description of the corresponding cluster.
- Each distribution in a mixture model is also called a **mixture component**.
- The distribution/component can be of any kind

AN EXAMPLE

- The figure shows a plot of the **probability density function** of a 1-dimensional data set (with two classes) generated by
  - a mixture of two Gaussian distributions,
  - one per class, whose parameters (denoted by θ_i) are the mean (μ_i) and the standard deviation (σ_i), i.e., $\theta_i = (\mu_i, \sigma_i)$.



MIXTURE MODEL (CONT …)

- Let the number of mixture components (or distributions) in a mixture model be *K*.
- Let the *j*th distribution have the parameters θ_j.
- Let Θ be the set of parameters of all components, $\Theta = \{\varphi_1, \varphi_2, \ldots, \varphi_K, \theta_1, \theta_2, \ldots, \theta_K\}$, where φ_j is the *mixture weight* (or *mixture probability*) of the mixture component *j* and θ_j is the parameters of component *j*.
- How does the model generate documents?

DOCUMENT GENERATION

- Due to one-to-one correspondence, each class corresponds to a mixture component. The mixture weights are *class prior probabilities*, i.e., $\varphi_j = \Pr(c_j|\Theta)$.
- The mixture model generates each document d_i by:
  - first selecting a mixture component (or class) according to class prior probabilities (i.e., mixture weights), $\varphi_j = \Pr(c_j|\Theta)$.
  - then having this selected mixture component (c_j) generate a document d_i according to its parameters, with distribution $\Pr(d_i|c_j; \Theta)$ or more precisely $\Pr(d_i|c_j; \theta_j)$.

$$\Pr(d_i \mid \Theta) = \sum_{j=1}^{|C|} \Pr(c_j \mid \Theta) \Pr(d_i \mid c_j; \Theta) \qquad (23)$$

MODEL TEXT DOCUMENTS

- The naïve Bayesian classification treats each document as a "bag of words". The generative model makes the following further assumptions:
  - Words of a document are generated independently of context given the class label. The familiar naïve Bayes assumption used before.
  - The probability of a word is independent of its position in the document. The document length is chosen independent of its class.

MULTINOMIAL DISTRIBUTION

- With the assumptions, each document can be regarded as generated by a multinomial distribution.
- In other words, each document is drawn from a multinomial distribution of words with as many independent trials as the length of the document.
- The words are from a given vocabulary $V = \{w_1, w_2, \ldots, w_{|V|}\}$.

USE PROBABILITY FUNCTION OF MULTINOMIAL DISTRIBUTION

$$\Pr(d_i \mid c_j;\Theta) = \Pr(|d_i|)\,|d_i|!\prod_{t=1}^{|V|}\frac{\Pr(w_t \mid c_j;\Theta)^{N_{ti}}}{N_{ti}!} \quad (24)$$

where N_{ti} is the number of times that word w_t occurs in document d_i and

$$\sum_{t=1}^{|V|} N_{it} = |d_i| \qquad \sum_{t=1}^{|V|}\Pr(w_t \mid c_j;\Theta) = 1. \quad (25)$$

PARAMETER ESTIMATION

- The parameters are estimated based on empirical counts.

$$\Pr(w_t \mid c_j; \hat{\Theta}) = \frac{\sum_{i=1}^{|D|} N_{ti} \Pr(c_j \mid d_i)}{\sum_{s=1}^{|V|} \sum_{i=1}^{|D|} N_{si} \Pr(c_j \mid d_i)}. \quad (26)$$

- In order to handle 0 counts for infrequent occurring words that do not appear in the training set, but may appear in the test set, we need to smooth the probability. *Lidstone* smoothing, $0 \leq \lambda \leq 1$

$$\Pr(w_t \mid c_j; \hat{\Theta}) = \frac{\lambda + \sum_{i=1}^{|D|} N_{ti} \Pr(c_j \mid d_i)}{\lambda |V| + \sum_{s=1}^{|V|} \sum_{i=1}^{|D|} N_{si} \Pr(c_j \mid d_i)}. \quad (27)$$

PARAMETER ESTIMATION (CONT …)

- Class prior probabilities, which are mixture weights φ_j, can be easily estimated using training data

$$\Pr(c_j \mid \hat{\Theta}) = \frac{\sum_{i=1}^{|D|} \Pr(c_j \mid d_i)}{|D|} \tag{28}$$

CLASSIFICATION

- Given a test document d_i, from

$$\Pr(c_j \mid d_i; \hat{\Theta}) = \frac{\Pr(c_j \mid \hat{\Theta}) \Pr(d_i \mid c_j; \hat{\Theta})}{\Pr(d_i \mid \hat{\Theta})}$$

$$= \frac{\Pr(c_j \mid \hat{\Theta}) \prod_{k=1}^{|d_i|} \Pr(w_{d_i,k} \mid c_j; \hat{\Theta})}{\sum_{r=1}^{|C|} \Pr(c_r \mid \hat{\Theta}) \prod_{k=1}^{|d_i|} \Pr(w_{d_i,k} \mid c_r; \hat{\Theta})}$$

where $w_{d_i,k}$ is the word in position k of document d_i. If the final classifier is to classify each document into a single class, then the class with the highest posterior probability is selected:

$$\arg\max_{c_j \in C} \Pr(c_j \mid d_i; \hat{\Theta}) \qquad (30)$$

DISCUSSIONS

- Most assumptions made by naïve Bayesian learning are violated to some degree in practice.
- Despite such violations, researchers have shown that naïve Bayesian learning produces very accurate models.
  - The main problem is the mixture model assumption. When this assumption is seriously violated, the classification performance can be poor.
- Naïve Bayesian learning is extremely efficient.

THANKS YOU