Naïve Bayes Classification

Dr. Wedad Hussein
wedad.hussein@cis.asu.edu.eg

Dr. Mahmoud Mounir
mahmoud.mounir@cis.asu.edu.eg

Data Mining:

Concepts and Techniques

(3rd ed.)

— Chapter 8 —

Jiawei Han, Micheline Kamber, and Jian Pei

University of Illinois at Urbana-Champaign &

Simon Fraser University

Chapter 8. Classification: Basic Concepts

- Classification: Basic Concepts
- Decision Tree Induction
- Bayes Classification Methods
- Model Evaluation and Selection
- Summary

Bayesian Classification: Why?

- A statistical classifier: performs *probabilistic prediction, i.e.,* predicts class membership probabilities
- Foundation: Based on Bayes' Theorem.
- Performance: A simple Bayesian classifier, *naïve Bayesian classifier*, has comparable performance with decision tree and selected neural network classifiers
- Incremental: Each training example can incrementally increase/decrease the probability that a hypothesis is correct — prior knowledge can be combined with observed data
- Standard: Even when Bayesian methods are computationally intractable, they can provide a standard of optimal decision making against which other methods can be measured

Bayes' Theorem: Basics

- Total probability Theorem: $P(B) = \sum_{i=1}^{M} P(B|A_i)P(A_i)$
- Bayes' Theorem: $$P(H|\mathbf{X}) = \frac{P(\mathbf{X}|H)P(H)}{P(\mathbf{X})} = P(\mathbf{X}|H) \times P(H) / P(\mathbf{X})$$
  - Let **X** be a data sample ("*evidence*"): class label is unknown
  - Let H be a *hypothesis* that X belongs to class C
  - Classification is to determine P(H|**X**), (i.e., *posteriori probability):* the probability that the hypothesis holds given the observed data sample **X**
  - P(H) (*prior probability*): the initial probability
    - E.g., **X** will buy computer, regardless of age, income, …
  - P(**X**): probability that sample data is observed
  - P(**X**|H) (likelihood): the probability of observing the sample **X**, given that the hypothesis holds
    - E.g., Given that **X** will buy computer, the prob. that X is 31..40, medium income

Prediction Based on Bayes' Theorem

- Given training data **X**, *posteriori probability of a hypothesis* H, P(H|**X**), follows the Bayes' theorem

$$P(H\,|\,\mathbf{X}) = \frac{P(\mathbf{X}|H)P(H)}{P(\mathbf{X})} = P(\mathbf{X}|H) \times P(H) / P(\mathbf{X})$$

- Informally, this can be viewed as

posteriori = likelihood x prior/evidence

- Predicts **X** belongs to C_i iff the probability $P(C_i|\mathbf{X})$ is the highest among all the $P(C_k|X)$ for all the *k* classes
- Practical difficulty: It requires initial knowledge of many probabilities, involving significant computational cost

Classification Is to Derive the Maximum Posteriori

- Let D be a training set of tuples and their associated class labels, and each tuple is represented by an n-D attribute vector $\mathbf{X} = (x_1, x_2, \ldots, x_n)$
- Suppose there are *m* classes $C_1, C_2, \ldots, C_m$.
- Classification is to derive the maximum posteriori, i.e., the maximal $P(C_i|\mathbf{X})$
- This can be derived from Bayes' theorem

$$P(C_i|\mathbf{X}) = \frac{P(\mathbf{X}|C_i)P(C_i)}{P(\mathbf{X})}$$

- Since P(X) is constant for all classes, only

$$P(C_i|\mathbf{X}) = P(\mathbf{X}|C_i)P(C_i)$$

needs to be maximized

Naïve Bayes Classifier

- A simplified assumption: attributes are conditionally independent (i.e., no dependence relation between attributes):

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

- This greatly reduces the computation cost: Only counts the class distribution
- If A_k is categorical, $P(x_k|C_i)$ is the # of tuples in C_i having value x_k for A_k divided by $|C_{i, D}|$ (# of tuples of C_i in D)
- If A_k is continous-valued, $P(x_k|C_i)$ is usually computed based on Gaussian distribution with a mean μ and standard deviation σ

and $P(x_k|C_i)$ is

$$g(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

$$P(\mathbf{X}|C_i) = g(x_k, \mu_{C_i}, \sigma_{C_i})$$

Naïve Bayes Classifier: Training Dataset

Example:

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

Class:

C1:buys_computer = ‘yes’ C2:buys_computer = ‘no’

Data to be classified:

X = (age <=30, Income = medium, Student = yes, Credit_rating = Fair)

Naïve Bayes Classifier: An Example

Class:

C1:buys_computer = 'yes'   **C2:buys_computer = 'no'**

- Compute $P(C_i)$ for each class:
  - **P(C1) = P(buys_computer = "yes") = 9/14 = 0.643**
  - **P(C2) = P(buys_computer = "no") = 5/14= 0.357**

Naïve Bayes Classifier: An Example

<u>Class:</u>

C1:buys_computer = 'yes' **C2:buys_computer = 'no'**

- Compute $P(X|C_i)$ for each class

$$P(X_k|C_1) = P(X_1|C_1) * P(X_2|C_1) * P(X_3|C_1)* \ldots.*P(X_k|C_1)$$

$$P(X_k|C_2) = P(X_1|C_2) * P(X_2|C_2) * P(X_3|C_2)* \ldots.*P(X_k|C_2)$$

Naïve Bayes Classifier: Training Dataset

Class:

C1:buys_computer = 'yes'   C2:buys_computer = 'no'

Data to be classified:

X = (age <=30, Income = medium, Student = yes, Credit_rating = Fair)

Age	Buys Computer	Count	Total	Conditional Probability	Conditional Probability
<= 30	**Yes**	**2**	**9**	**(2/9)**	**0.222222222**
<= 30	**No**	**3**	**5**	**(3/5)**	**0.6**
31-40	Yes	4	9	(4/9)	0.444444444
31-40	No	0	5	(0/5)	0
> 40	Yes	3	9	(3/9)	0.333333333
> 40	No	2	5	(2/5)	0.4

P(Age <= 30| Buys Computer = Yes)	**0.222222222**
P(Age <= 30| Buys Computer = No)	**0.6**
P(Age Between 31 and 40| Buys Computer = Yes)	0.444444444
P(Age Between 31 and 40| Buys Computer = No)	0
P(Age > 40| Buys Computer = Yes)	0.333333333
P(Age > 40| Buys Computer = No)	0.4

Naïve Bayes Classifier: Training Dataset

Class:

C1:buys_computer = 'yes' C2:buys_computer = 'no'

Data to be classified:

X = (age <=30, Income = medium, Student = yes, Credit_rating = Fair)

Income	Buys Computer	Count	Total	Conditional Probability	Conditional Probability
High	Yes	2	9	(2/9)	0.222222222
High	No	2	5	(2/5)	0.4
Medium	**Yes**	**4**	**9**	**(4/9)**	**0.444444444**
Medium	**No**	**2**	**5**	**(2/5)**	**0.4**
Low	Yes	3	9	(3/9)	0.333333333
Low	No	1	5	(1/5)	0.2

P(Income = High| Buys Computer = Yes)	0.222222222
P(Income = High| Buys Computer = No)	0.4
P(Income = Medium| Buys Computer = Yes)	**0.444444444**
P(Income = Medium| Buys Computer = No)	**0.4**
P(Income = Low| Buys Computer = Yes)	0.333333333
P(Income = Low| Buys Computer = No)	0.2

Naïve Bayes Classifier: Training Dataset

Class:

C1:buys_computer = 'yes'　　C2:buys_computer = 'no'

Data to be classified:

X = (age <=30, Income = medium, Student = yes, Credit_rating = Fair)

Student	Buys Computer	Count	Total	Conditional Probability	Conditional Probability
Yes	**Yes**	**6**	**9**	**(6/9)**	**0.666666667**
Yes	**No**	**1**	**5**	**(1/5)**	**0.2**
No	Yes	3	9	(3/9)	0.333333333
No	No	4	5	(4/5)	0.8

P(Student = Yes| Buys Computer = Yes)	**0.666666667**
P(Student = Yes| Buys Computer = No)	**0.2**
P(Student = No| Buys Computer = Yes)	0.333333333
P(Student = No| Buys Computer = No)	0.8

Naïve Bayes Classifier: Training Dataset

Class:

C1:buys_computer = 'yes' C2:buys_computer = 'no'

Data to be classified:

X = (age <=30, Income = medium, Student = yes, Credit_rating = Fair)

Credit Rating	Buys Computer	Count	Total	Conditional Probability	Conditional Probability
Fair	**Yes**	**6**	**9**	**(6/9)**	**0.666666667**
Fair	**No**	**2**	**5**	**(2/5)**	**0.4**
Excellent	Yes	3	9	(3/9)	0.333333333
Excellent	No	3	5	(3/5)	0.6

P(Credit Rating = Fair| Buys Computer = Yes)	**0.666666667**
P(Credit Rating = Fair| Buys Computer = No)	**0.4**
P(Credit Rating = Excellent| Buys Computer = Yes)	0.333333333
P(Credit Rating = Excellent| Buys Computer = No)	0.6

Naïve Bayes Classifier: An Example

Class:

C1:buys_computer = 'yes' **C2:buys_computer = 'no'**

- Compute $P(X|C_i)$ for each class

$P(X|C_1) = P(X|buys\_computer = \text{"yes"})$

$= 0.222 \times 0.444 \times 0.667 \times 0.667 = 0.044$

$P(X|C_2) = P(X|buys\_computer = \text{"no"})$

$= 0.6 \times 0.4 \times 0.2 \times 0.4 = 0.019$

Naïve Bayes Classifier: An Example

<u>Class:</u>

C1:buys_computer = 'yes' **C2:buys_computer = 'no'**

- Compute $P(X|C_i) * P(C_i)$ for each class

  $$P(X|C_1) * P(C_1) = 0.044 * 0.643 = 0.028$$

  $$P(X|C_2) * P(C_2) = 0.019 * 0.357 = 0.007$$

- **Decision**

  $$P(X|C_1) * P(C_1) > P(X|C_2) * P(C_2)$$

<u>X belongs to (C_1)</u>

<u>Therefore, X belongs to class ("buys_computer = yes")</u>

Naïve Bayes Classifier: An Example

- $P(C_i)$: P(buys_computer = "yes") = 9/14 = 0.643
  P(buys_computer = "no") = 5/14= 0.357
- Compute $P(X|C_i)$ for each class
  P(age = "<=30" | buys_computer = "yes") = 2/9 = 0.222
  P(age = "<= 30" | buys_computer = "no") = 3/5 = 0.6
  P(income = "medium" | buys_computer = "yes") = 4/9 = 0.444
  P(income = "medium" | buys_computer = "no") = 2/5 = 0.4
  P(student = "yes" | buys_computer = "yes) = 6/9 = 0.667
  P(student = "yes" | buys_computer = "no") = 1/5 = 0.2
  P(credit_rating = "fair" | buys_computer = "yes") = 6/9 = 0.667
  P(credit_rating = "fair" | buys_computer = "no") = 2/5 = 0.4
- **X = (age <= 30 , income = medium, student = yes, credit_rating = fair)**

$P(X|C_i)$: P(X|buys_computer = "yes") = 0.222 x 0.444 x 0.667 x 0.667 = 0.044
P(X|buys_computer = "no") = 0.6 x 0.4 x 0.2 x 0.4 = 0.019

$P(X|C_i)*P(C_i)$: P(X|buys_computer = "yes") * P(buys_computer = "yes") = 0.028
P(X|buys_computer = "no") * P(buys_computer = "no") = 0.007

Therefore, X belongs to class ("buys_computer = yes")

Avoiding the Zero-Probability Problem

- Naïve Bayesian prediction requires each conditional prob. be **non-zero**. Otherwise, the predicted prob. will be zero

$$P(X|C_i) = \prod_{k=1}^{n} P(x_k|C_i)$$

- Ex. Suppose a dataset with 1000 tuples, income=low (0), income= medium (990), and income = high (10)
- Use **Laplacian correction** (or Laplacian estimator)
  - *Adding 1 to each case*

    Prob(income = low) = 1/1003

    Prob(income = medium) = 991/1003

    Prob(income = high) = 11/1003
  - The "corrected" prob. estimates are close to their "uncorrected" counterparts

Naïve Bayes Classifier: Comments

- Advantages
  - Easy to implement
  - Good results obtained in most of the cases
- Disadvantages
  - Assumption: class conditional independence, therefore loss of accuracy
  - Practically, dependencies exist among variables
    - E.g., hospitals: patients: Profile: age, family history, etc.

      Symptoms: fever, cough etc., Disease: lung cancer, diabetes, etc.
    - Dependencies among these cannot be modeled by Naïve Bayes Classifier
- How to deal with these dependencies? Bayesian Belief Networks (Chapter 9)

(c) Consider the use of real-valued attributes, when learning decision trees, as described in the lecture. The table below shows the relationship between the body height and the gender of a group of persons (the records have been sorted with respect to the value of *height* in cm). Calculate the information gain for potential splitting thresholds and determine the best one.

Height	161	164	169	175	176	179	180	184	185
Gender	F	F	M	M	F	F	M	M	F

(c) Consider the use of real-valued attributes, when learning decision trees, as described in the lecture. The table below shows the relationship between the body height and the gender of a group of persons (the records have been sorted with respect to the value of *height* in cm). Calculate the information gain for potential splitting thresholds and determine the best one.

Height	161	164	169	175	176	179	180	184	185
Gender	F	F	M	M	F	F	M	M	F

- Potential cut points must lie in the intervals $(164, 169)$, $(175, 176)$, $(179, 180)$, or $(184, 185)$.

- Calculate the information gain for the potential splitting thresholds
- $C_1 \in (164, 169)$
  - resulting class distribution: if $x < C_1$ then $2 - 0$ else $3 - 4$
  - conditional entropy: if $x < C_1$ then $E = 0$ else $E = -\frac{3}{7}\log_2\frac{3}{7} - \frac{4}{7}\log_2\frac{4}{7} = 0.985$
  - entropy: $E(C_1|S) = \frac{2}{9}\cdot 0 + \frac{7}{9}\cdot 0.985 = 0.766$
- $C_2 \in (175, 176)$
  - resulting class distribution: if $x < C_2$ then $2 - 2$ else $3 - 2$
  - entropy: $E(C_2|S) = \frac{4}{9}\cdot 1 + \frac{5}{9}\cdot 0.971 = 0.984$
- $C_3 \in (179, 180)$
  - resulting class distribution: if $x < C_3$ then $4 - 2$ else $1 - 2$
  - entropy: $E(C_3|S) = \frac{6}{9}\cdot 0.918 + \frac{3}{9}\cdot 0.918 = 0.918$
- $C_4 \in (184, 185)$
  - resulting class distribution: if $x < C_4$ then $4 - 4$ else $1 - 0$
  - entropy: $E(C_4|S) = \frac{8}{9}\cdot 1 + \frac{1}{9}\cdot 0 = 0.889$

References (1)

- C. Apte and S. Weiss. **Data mining with decision trees and decision rules**. Future Generation Computer Systems, 13, 1997
- C. M. Bishop, **Neural Networks for Pattern Recognition**. Oxford University Press, 1995
- L. Breiman, J. Friedman, R. Olshen, and C. Stone. **Classification and Regression Trees**. Wadsworth International Group, 1984
- C. J. C. Burges. **A Tutorial on Support Vector Machines for Pattern Recognition**. *Data Mining and Knowledge Discovery*, 2(2): 121-168, 1998
- P. K. Chan and S. J. Stolfo. **Learning arbiter and combiner trees from partitioned data for scaling machine learning**. KDD'95
- H. Cheng, X. Yan, J. Han, and C.-W. Hsu, **Discriminative Frequent Pattern Analysis for Effective Classification**, ICDE'07
- H. Cheng, X. Yan, J. Han, and P. S. Yu, **Direct Discriminative Pattern Mining for Effective Classification**, ICDE'08
- W. Cohen. **Fast effective rule induction**. ICML'95
- G. Cong, K.-L. Tan, A. K. H. Tung, and X. Xu. **Mining top-k covering rule groups for gene expression data**. SIGMOD'05

References (2)

- A. J. Dobson. **An Introduction to Generalized Linear Models**. Chapman & Hall, 1990.
- G. Dong and J. Li. **Efficient mining of emerging patterns: Discovering trends and differences**. KDD'99.
- R. O. Duda, P. E. Hart, and D. G. Stork. **Pattern Classification**, 2ed. John Wiley, 2001
- U. M. Fayyad. **Branching on attribute values in decision tree generation**. AAAI'94.
- Y. Freund and R. E. Schapire. **A decision-theoretic generalization of on-line learning and an application to boosting**. J. Computer and System Sciences, 1997.
- J. Gehrke, R. Ramakrishnan, and V. Ganti. **Rainforest: A framework for fast decision tree construction of large datasets**. VLDB'98.
- J. Gehrke, V. Gant, R. Ramakrishnan, and W.-Y. Loh, **BOAT -- Optimistic Decision Tree Construction**. SIGMOD'99.
- T. Hastie, R. Tibshirani, and J. Friedman. **The Elements of Statistical Learning: Data Mining, Inference, and Prediction.** Springer-Verlag, 2001.
- D. Heckerman, D. Geiger, and D. M. Chickering. **Learning Bayesian networks: The combination of knowledge and statistical data**. Machine Learning, 1995.
- W. Li, J. Han, and J. Pei, **CMAR: Accurate and Efficient Classification Based on Multiple Class-Association Rules**, ICDM'01.

References (3)

- T.-S. Lim, W.-Y. Loh, and Y.-S. Shih. **A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms.** Machine Learning, 2000.
- J. Magidson. **The Chaid approach to segmentation modeling: Chi-squared automatic interaction detection**. In R. P. Bagozzi, editor, Advanced Methods of Marketing Research, Blackwell Business, 1994.
- M. Mehta, R. Agrawal, and J. Rissanen. **SLIQ : A fast scalable classifier for data mining**. EDBT'96.
- T. M. Mitchell. **Machine Learning**. McGraw Hill, 1997.
- S. K. Murthy, **Automatic Construction of Decision Trees from Data: A Multi-Disciplinary Survey**, Data Mining and Knowledge Discovery 2(4): 345-389, 1998
- J. R. Quinlan. **Induction of decision trees**. *Machine Learning*, 1:81-106, 1986.
- J. R. Quinlan and R. M. Cameron-Jones. **FOIL: A midterm report**. ECML'93.
- J. R. Quinlan. **C4.5: Programs for Machine Learning**. Morgan Kaufmann, 1993.
- J. R. Quinlan. **Bagging, boosting, and c4.5**. AAAI'96.

References (4)

- R. Rastogi and K. Shim. **Public: A decision tree classifier that integrates building and pruning**. VLDB’98.
- J. Shafer, R. Agrawal, and M. Mehta. **SPRINT : A scalable parallel classifier for data mining**. VLDB’96.
- J. W. Shavlik and T. G. Dietterich. **Readings in Machine Learning**. Morgan Kaufmann, 1990.
- P. Tan, M. Steinbach, and V. Kumar. **Introduction to Data Mining**. Addison Wesley, 2005.
- S. M. Weiss and C. A. Kulikowski. **Computer Systems that Learn: Classification and Prediction Methods from Statistics, Neural Nets, Machine Learning, and Expert Systems**. Morgan Kaufman, 1991.
- S. M. Weiss and N. Indurkhya. **Predictive Data Mining**. Morgan Kaufmann, 1997.
- I. H. Witten and E. Frank. **Data Mining: Practical Machine Learning Tools and Techniques**, 2ed. Morgan Kaufmann, 2005.
- X. Yin and J. Han. **CPAR: Classification based on predictive association rules**. SDM'03
- H. Yu, J. Yang, and J. Han. **Classifying large data sets using SVM with hierarchical clusters**. KDD'03.