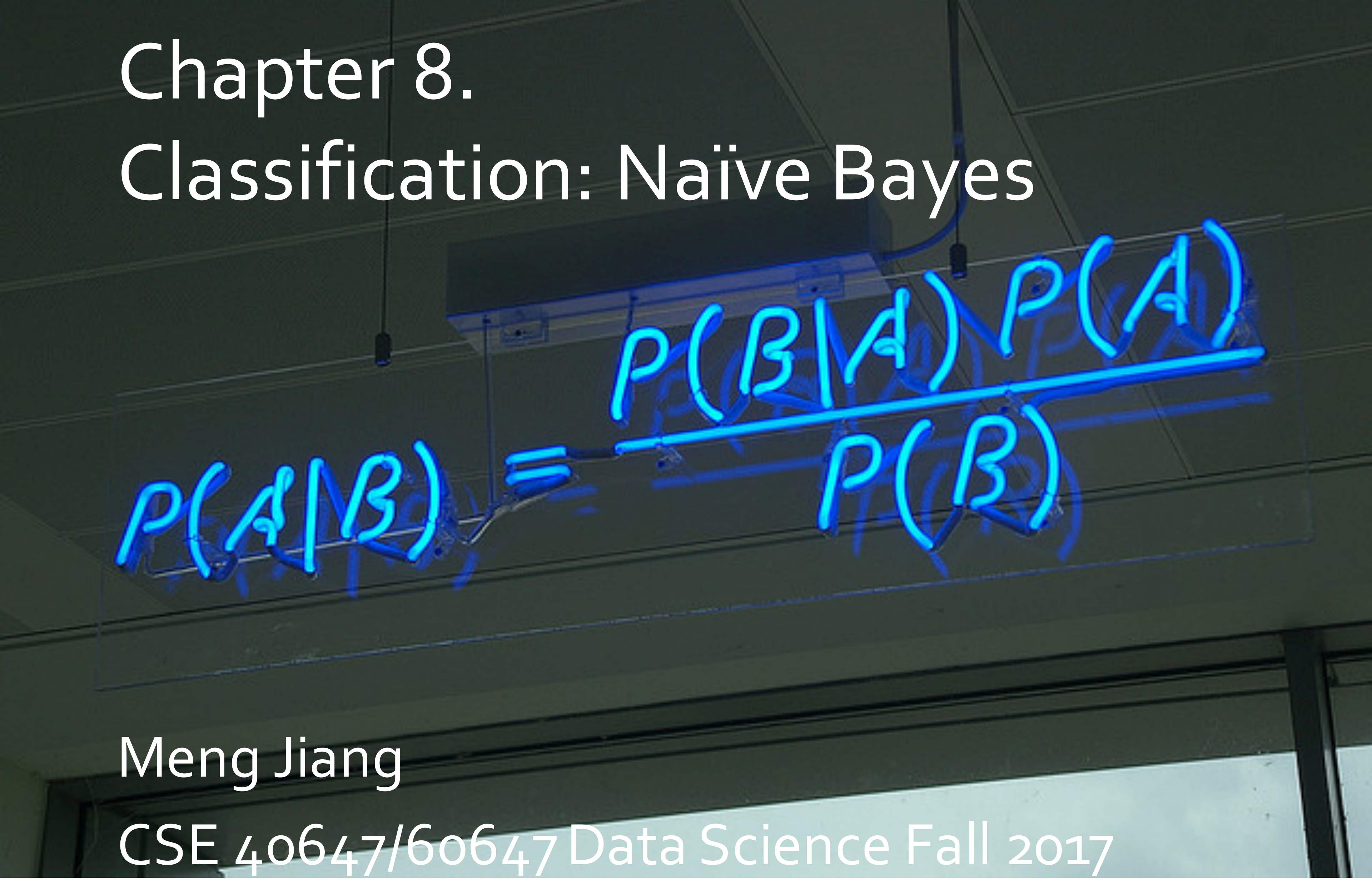


Chapter 8.
Classification: Naïve Bayes

Meng Jiang

CSE 40647/60647 Data Science Fall 2017

Introduction to Data Mining

Bayes' Theorem: Basics

Proof of Bayes Theorem

The probability of two events A and B happening, P(A∩B), is the probability of A, P(A), times the probability of B given that A has occurred, P(B|A).

$$P(A \cap B) = P(A)P(B|A) \tag{1}$$

On the other hand, the probability of A and B is also equal to the probability of B times the probability of A given B.

$$P(A \cap B) = P(B)P(A|B) \tag{2}$$

Equating the two yields:

$$P(B)P(A|B) = P(A)P(B|A) \tag{3}$$

and thus

$$\boxed{P(A|B) = P(A)\frac{P(B|A)}{P(B)}} \tag{4}$$

This equation, known as Bayes Theorem is the basis of statistical inference.

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

Bayes' Theorem: Basics

- Bayes' Theorem:
  - Let **X** be a data sample: 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
  - P(**X**) (*evidence*): probability that sample data is observed
  - P(**X**|H) (*likelihood*): the probability of observing the sample **X**, given that the hypothesis holds

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

Prediction Based on Bayes' Theorem

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

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

- 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 $P(C_k|\mathbf{X})$ 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 $P(\mathbf{X})$ 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 data D)
- If A_k is continuous-valued, $P(x_k|C_i)$ is usually computed based on Gaussian distribution with a mean μ and standard deviation σ

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

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

Quinlan’s Example (1986): Playing Tennis

	Outlook	Temperature	Humidity	Windy	Label: Play?
1	Sunny	Hot	High	"False"	No
2	Sunny	Hot	High	"True"	No
3	Overcast	Hot	High	"False"	Yes
4	Rainy	Mild	High	"False"	Yes
5	Rainy	Cool	Normal	"False"	Yes
6	Rainy	Cool	Normal	"True"	No
7	Overcast	Cool	Normal	"True"	Yes
8	Sunny	Mild	High	"False"	No
9	Sunny	Cool	Normal	"False"	Yes
10	Rainy	Mild	Normal	"False"	Yes
11	Sunny	Mild	Normal	"True"	Yes
12	Overcast	Mild	High	"True"	Yes
13	Overcast	Hot	Normal	"False"	Yes
14	Rainy	Mild	High	"True"	No
1	**Rainy**	**Hot**	**High**	**"False"**	**?**

P(H): Prior Probability

$P(C_i)$

- P(Play? = “yes”) = 9/14 = 0.643
- P(Play? = “no”) = 5/14= 0.357

Quinlan's Example (1986): Playing Tennis

	Outlook	Temperature	Humidity	Windy	Label: Play?
1	Sunny	Hot	High	"False"	No
2	Sunny	Hot	High	"True"	No
3	Overcast	Hot	High	"False"	Yes
4	Rainy	Mild	High	"False"	Yes
5	Rainy	Cool	Normal	"False"	Yes
6	Rainy	Cool	Normal	"True"	No
7	Overcast	Cool	Normal	"True"	Yes
8	Sunny	Mild	High	"False"	No
9	Sunny	Cool	Normal	"False"	Yes
10	Rainy	Mild	Normal	"False"	Yes
11	Sunny	Mild	Normal	"True"	Yes
12	Overcast	Mild	High	"True"	Yes
13	Overcast	Hot	Normal	"False"	Yes
14	Rainy	Mild	High	"True"	No
1	Rainy	Hot	High	"False"	?

$P(\mathbf{X}|H)$: Likelihood

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

- P(Outlook = Rainy | Play? = “yes”) = 3/9 = 0.333
- P(Outlook = Rainy | Play? = “no”) = 2/5 = 0.4

- P(Temperature = Hot | Play? = “yes) = 2/9 = 0.222
- P(Temperature = Hot | Play? = “no”) = 2/5 = 0.4

- P(Humidity = High | Play? = “yes”) = 3/9 = 0.333
- P(Humidity = High | Play? = “no”) = 4/5 = 0.8

- P(Windy = “False” | Play? = “yes”) = 6/9 = 0.667
- P(Windy= “False” | Play? = “no”) = 2/5 = 0.4

$P(H|\mathbf{X})$: Posteriori Probability

$\mathbf{X}$ = (Outlook=Rainy, Temperature=Hot, Humidity=High, Windy="False")

$P(\mathbf{X}) = (5/14) \times (4/14) \times (7/14) \times (8/14) = 0.02915$

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

$P(\mathbf{X} \mid \text{Play?} = \text{"yes"}) = 0.333 \times 0.222 \times 0.333 \times 0.667 = 0.01642$

$P(\mathbf{X} \mid \text{Play?} = \text{"no"}) = 0.4 \times 0.4 \times 0.8 \times 0.4 = 0.0512$

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

$P(\text{Play?} = \text{"yes"} \mid \mathbf{X}) = P(\mathbf{X} \mid \text{Play?} = \text{"yes"}) * P(\text{Play?} = \text{"yes"}) / P(\mathbf{X})$

$= 0.01642 \times 0.643 / 0.02915 = 0.36$

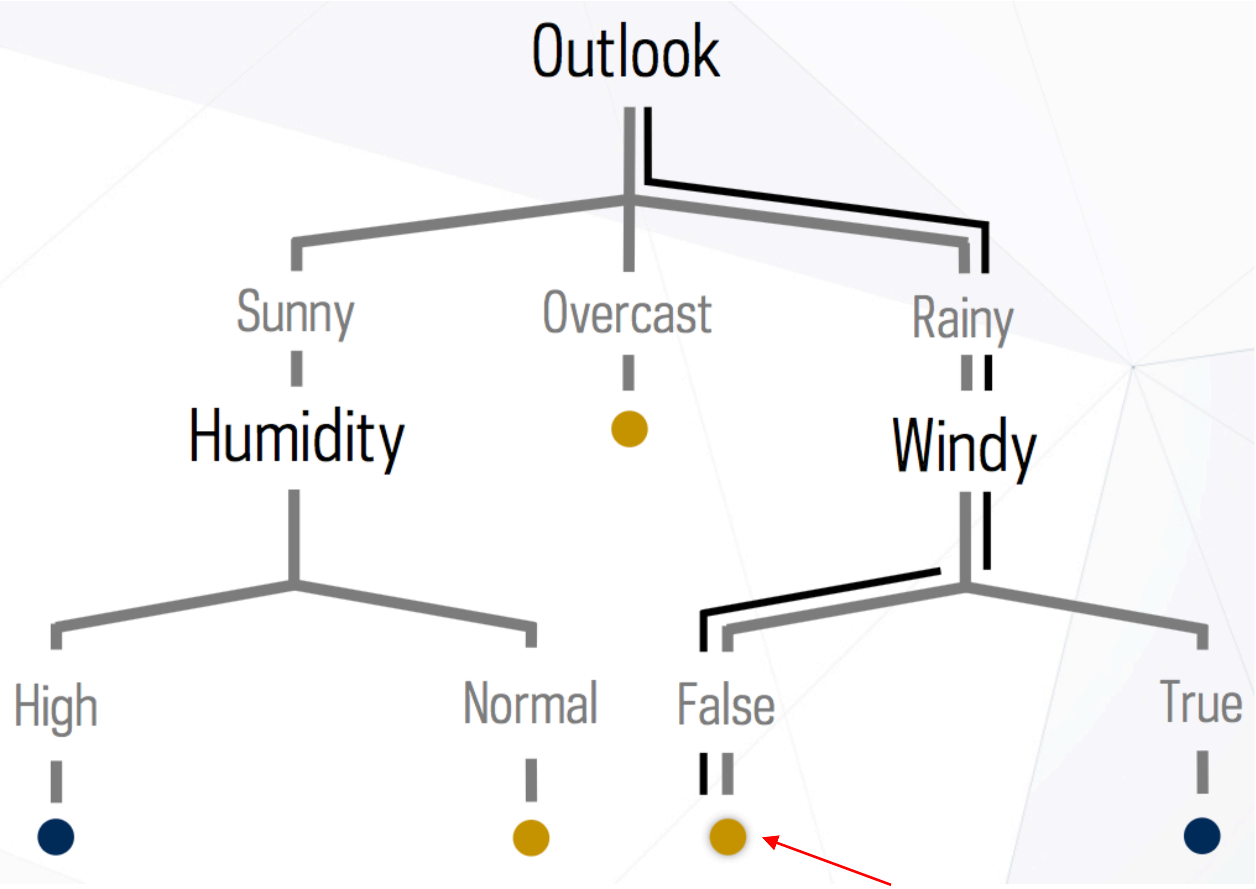
$P(\text{Play?} = \text{"no"} \mid \mathbf{X}) = P(\mathbf{X} \mid \text{Play?} = \text{"no"}) * P(\text{Play?} = \text{"no"}) / P(\mathbf{X})$

$= 0.0512 \times 0.357 / 0.02915 = 0.63$

So, the conclusion is *Play? = "no".*

Call Back: Decision Tree-Prediction

1	Rainy	Hot	High	"False"	?



"Yes"

Quinlan's Example (1986): Playing Tennis

	Outlook	Temperature	Humidity	Windy	Label: Play?
1	Sunny	Hot	High	"False"	No
2	Sunny	Hot	High	"True"	No
3	Overcast	Hot	High	"False"	Yes
4	Rainy	Mild	High	"False"	Yes
5	Rainy	Cool	Normal	"False"	Yes
6	Rainy	Cool	Normal	"True"	No
7	Overcast	Cool	Normal	"True"	Yes
8	Sunny	Mild	High	"False"	No
9	Sunny	Cool	Normal	"False"	Yes
10	Rainy	Mild	Normal	"False"	Yes
11	Sunny	Mild	Normal	"True"	Yes
12	Overcast	Mild	High	"True"	Yes
13	Overcast	Hot	Normal	"False"	Yes
14	Rainy	Mild	High	"True"	No
1	**Rainy**	**Hot**	**High**	**"False"**	**?**

Avoiding the Zero-Probability Problem

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

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

- 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., Hospital-patient data
      - Patient profile: age, family history, etc.
      - Symptoms: fever, cough, etc.
      - Disease: lung cancer, diabetes, etc.
    - Dependencies among these cannot be modeled by Naïve Bayes Classifier

References

- C. Apte and S. Weiss. Data mining with decision trees and decision rules. Future Generation Computer Systems, 13, 1997
- P. K. Chan and S. J. Stolfo. Learning arbiter and combiner trees from partitioned data for scaling machine learning. KDD'95
- A. J. Dobson. An Introduction to Generalized Linear Models. Chapman & Hall, 1990.
- 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.
- 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

References (cont.)

- 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. C4.5: Programs for Machine Learning. Morgan Kaufmann, 1993.
- J. R. Quinlan. Bagging, boosting, and c4.5. AAAI'96.
- 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