Bayes Classifier and Bayesian Networks

Outline

- Bayes Classifier
- Bayesian Networks

Bayes Classifier

What Is Bayes Classifier?

- A statistical classifier
  - Perform *probabilistic prediction* (*i.e.,* predict the probability of a class membership)
- Foundation - Based on Bayes' Theorem
- Performance: A simple Bayes classifier, *naïve Bayes classifier*, has comparable performance with decision tree and many other classification methods
- Incremental
  - Each training example can incrementally increase/decrease the probability that a hypothesis is correct: Prior knowledge can be combined with observed data
- **Theoretical 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:

$$p(B) = \sum_{i} p\,(B|A_i)p(A_i)$$

- Bayes' Theorem:

$$\boxed{p(H|\mathbf{X})} = \frac{p(\mathbf{X}|H)P(H)}{p(\mathbf{X})} \propto \boxed{p(\mathbf{X}|H)}\,\boxed{P(H)}$$

posteriori probability — What we should choose

likelihood — What we just see

prior probability — What we knew previously

- **X:** A data sample ("*evidence*")
- H: X belongs to class C

Prediction can be done based on Bayes' Theorem:

Classification is to derive the maximum posteriori

Naïve Bayes Classifier: Making a Naïve Assumption

- Practical difficulty of Bayes classifier: It requires initial knowledge of many probabilities, which may not be available or involving significant computational cost
- A Naïve special case
  - Make an additional assumption to simplify the model, but achieve comparable performance

Attributes are conditionally independent
(i.e., no dependence relation between attributes)

$$p(X|C_i) = \prod_k p(x_k|C_i) \boxed{= p(x_1|C_i) \cdot p(x_2|C_i) \cdots\cdots p(x_n|C_i)}$$

  - Only need to count the class distribution w.r.t. features
- Naive Bayes classifier: Combines the independent feature model with a decision rule
  - The *MAP* (*maximum a posteriori*) decision rule: Pick the hypothesis that is most probable

Naïve Bayes Classifier: Categorical vs. Continuous Valued Features

- If feature $\mathrm{x_k}$ is categorical, $\mathrm{p(x_k = v_k|C_i)}$ is the # of tuples in $\mathrm{C_i}$ with $\mathrm{x_k = v_k}$, divided by $|C_{i,\,D}|$ (# of tuples of C_i in D)

$$\mathrm{p}(\mathrm{X}|C_i) = \prod_{\mathrm{k}} \mathrm{p(x_k|C_i) = p(x_1|C_i) \cdot p(x_2|C_i) \cdots\cdots p(x_n|C_i)}$$

- If feature $\mathrm{x_k}$ is continuous-valued, $\mathrm{p(x_k = v_k|C_i)}$ is usually computed based on Gaussian distribution with a mean μ and standard deviation σ

$$\mathrm{p(x_k = v_k|C_i)} = N\left(\mathrm{x_k}\middle|\mathrm{\mu_{C_i}, \sigma_{C_i}}\right) = \frac{1}{\sqrt{2\pi}\sigma_{\mathrm{C_i}}} e^{-\frac{\left(x-\mu_{C_i}\right)^2}{2\sigma^2}}$$

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	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

Naïve Bayes Classifier: An 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

- $P(C_i)$: 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 probability be **non-zero**
  - Otherwise, the predicted probability will be zero

$$\mathrm{p}(\mathrm{X}|C_i) = \prod_k p(x_k|C_i) = p(x_1|C_i) \cdot p(x_2|C_i) \cdots\cdots p(x_n|C_i)$$

- Example. 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/(1000 + 3)

    Prob(income = medium) = (990 + 1)/(1000 + 3)

    Prob(income = high) = (10 + 1)/(1000 + 3)
  - The "corrected" probability estimates are close to their "uncorrected" counterparts

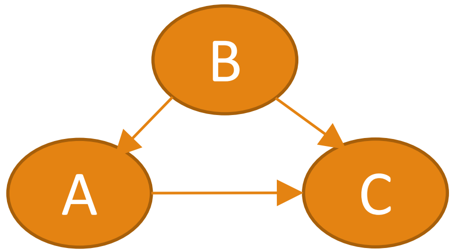
Naïve Bayes Classifier: Strength vs. Weakness

- Strength
  - Easy to implement
  - Good results obtained in most of the cases
- Weakness
  - Assumption: Attributes conditional independence, therefore loss of accuracy
  - Practically, dependencies exist among variables
    - E.g., 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?
  - Use Bayesian Belief Networks

Bayesian Networks

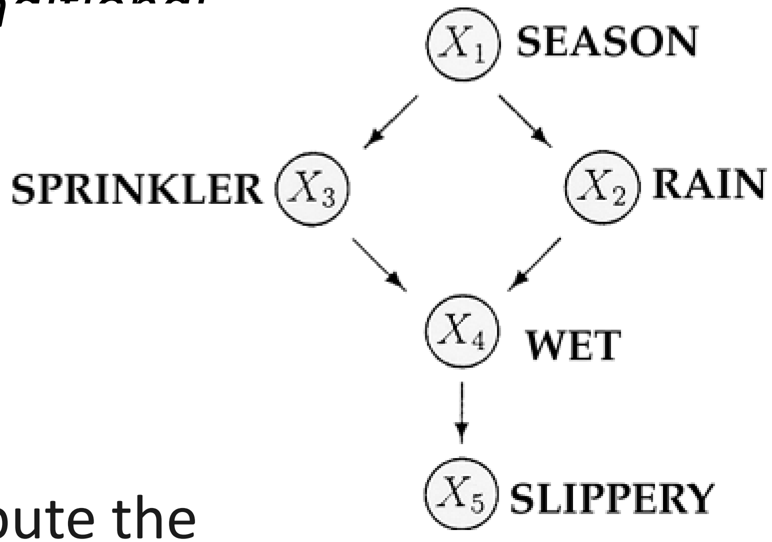
From Naïve Bayes to Bayesian Network

- A naïve Bayes classifier assumes that the value of a particular feature is independent of the value of any other feature, given the class variable
  - This assumption is often too simple to model the real world well
- Bayesian network (or Bayes network, belief network, Bayesian model, or probabilistic directed acyclic graphical model) is a probabilistic **graphical model**
  - Represented by a set of *random variables* and *their conditional dependencies* via a *directed acyclic graph* (DAG)



$p(A, B, C) = p(B) \cdot p(A|B) \cdot p(C|A, B)$

directed **acyclic** graphical model

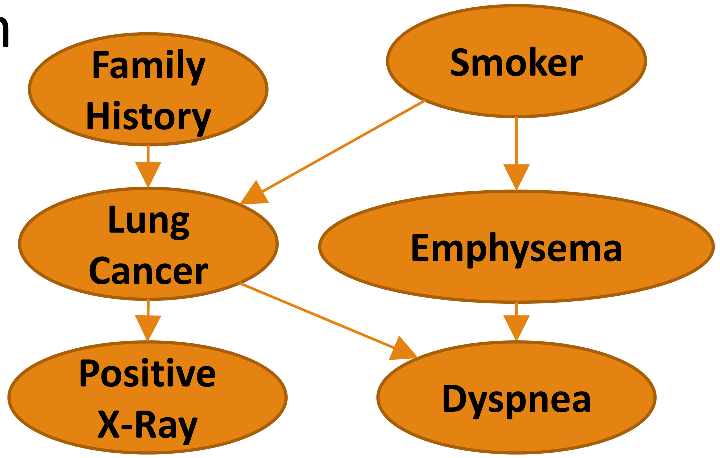


  - Ex. Given symptoms, the network can be used to compute the probabilities of the presence of various diseases

Bayesian Network

- **Bayesian network** (or **Bayesian belief network, probabilistic network**):
  - Allows *class conditional independencies* between *subsets* of variables
  - Represents the joint distribution compactly in a factorized way
  - Can be described by a generative process
- Two components:
  - A *directed acyclic graph* (called a structure)
    - The **nodes** represent random variables, observed or hidden
    - The **edges** represent direct dependency between variables
  - A set of *conditional probability tables* (CPTs)
    - CPTs are attached to the nodes

Nodes: random variables   Links: dependency



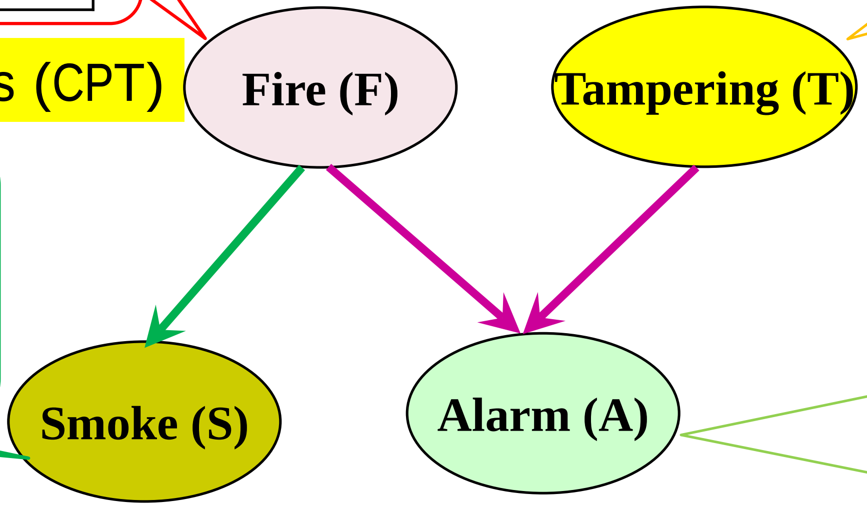
	FH, S	$FH, \sim S$	$\sim FH, S$	$\sim FH, \sim S$
LC	0.8	0.5	0.7	0.1
$\sim LC$	0.2	0.5	0.3	0.9

Representing Joint Distribution with Bayesian Network

Fire=1	Fire=0
0.02	0.98

Tampering=1	Tampering=0
0.1	0.9

Conditional **P**robability **T**ables (CPT)



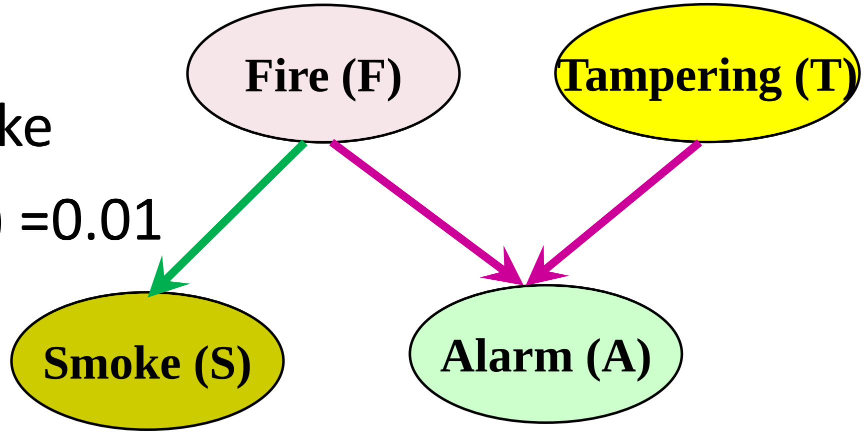
	Smoke=1	Smoke=0
Fire=1	0.9	0.1
Fire=0	0.01	0.99

	Alarm=1	Alarm=0
F=1,T=1	0.5	0.5
F=1,T=0	0.99	0.01
F=0,T=1	0.85	0.15
F=0,T=0	0.0001	0.9999

From the Bayesian network we can read the factorization of the joint distribution: $p(F, S, A, T) = p(F) \cdot p(T) \cdot p(S|F) \cdot p(A|F, T)$

In general, we have the chain rule for Bayesian networks: $p(X) = \prod_k p(x_k|\text{Parents}(x_k))$

Bayesian Network: An Example

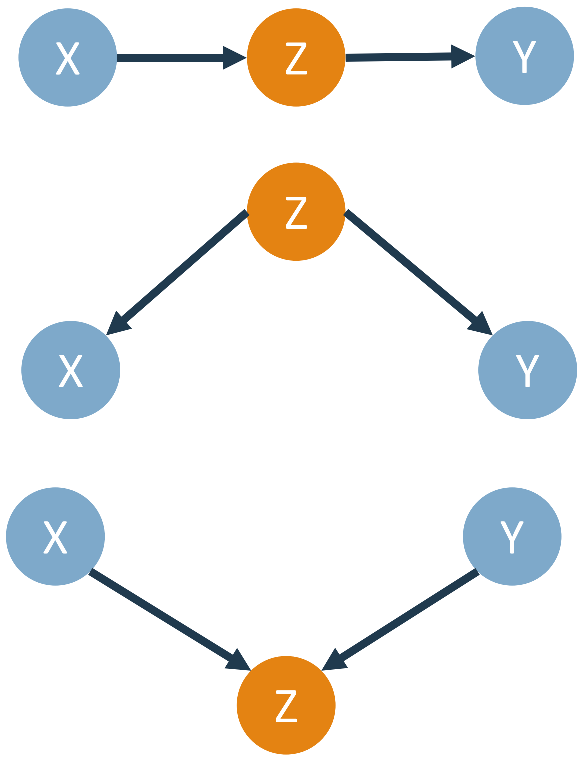


- Causal Reasoning:
  - The value of variable Fire influences variable Smoke
    - If F=1 → p(S=1|F=1) =0.9; if F=0 → p(S=1|F=0) =0.01
- Evidential Reasoning:
  - The value of variable Smoke also influences Fire
    - If S=1 → p(F=1|S=1) = $\frac{p(S=1|F=1)*p(F=1)}{p(S=1)}$ = 0.647; if S=0, p(F=1|S=0)=0.002
- Intercausal Reasoning:
  - The value of variable Fire does not influence Tampering
    - p(T|F) = p(T), F and T are independent
  - However, observing Alarm makes Fire and Tampering coupled
    - p(T|F, A) = $\frac{p(T,A,F)}{p(A|F)p(F)} = \frac{p(A|F,T)p(F)p(T)}{p(F)\sum_T p(A,T|F)} = \frac{p(A|F,T)p(T)}{\sum_T p(A|F,T)p(T)}$
    - p(T=1|F=1, A=1) = 0.053; p(T=1|F=0, A=1) ≈ 1.000

Dependencies in Bayesian Network

- Markovian assumption
  - Each variable becomes independent of its non-effects once its direct causes are known

- Consider the local structure of three variables
  - Cascade
    - Given the middle node Z, X, and Y are independent
  - Common parent
    - Given the parent node Z, X, and Y are independent
  - V-structure (common child):
    - Given the child node Z, X, and Y are coupled
      - Also called X explains away Y w.r.t. Z



Three Major Tasks on Bayesian Network

- **Representation (Construction)**
  - How can we construct a Bayesian network so that the probability distribution reflects real-world phenomenon?
- **Inference**
  - Given a Bayesian network, how can we answer questions about events?
  - Marginal inference: What is the probability of a variable after summing all other variables?
  - MAP inference: What is the most likely assignment to the unobserved variables?
- **Learning (Training)**
  - How can we fit a Bayesian network to data?
  - In some cases, the network structure needs to be learned as well

How Are Bayesian Networks Constructed?

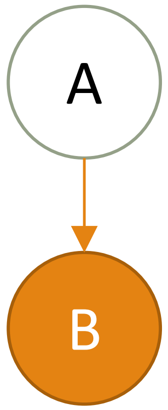
- **Subjective construction**
  - Human identification of (direct) causal structure
  - People are quite good at identifying direct causes from a given set of variables & whether the set contains all relevant direct causes
- **Synthesis from other specifications**
  - E.g., from a formal system design: Block diagrams & information flow
- **Learning from data** (e.g., from medical records or student admission records)
  - Learn parameters given its structure or learn both structure and parameters
  - Maximum likelihood principle: Favors Bayesian networks that maximize the probability of observing the given data set

Training Bayesian Networks: Several Scenarios

- Scenario 1: Given both the network structure and all variables observable
  - *Compute only the CPT entries*
- Scenario 2: Network structure known, some variables hidden
  - Use *gradient descent* (a greedy hill-climbing method), i.e., search for a solution along the steepest descent of a criterion function
- Scenario 3: Network structure unknown, all variables observable
  - Search through the model space to *reconstruct network topology*
- Scenario 4: Unknown structure, all hidden variables
  - No good algorithms known for this purpose
- D. Heckerman. A Tutorial on Learning with Bayesian Networks. In *Learning in Graphical Models,* M. Jordan, ed. MIT Press, 1999

Probabilistic Graphic Model: Plate Notation

- How to represent variables and their dependencies concisely in a graphical model?
- Node (or **circle**) representing variables
  - A solid (or shaded) circle means the corresponding variable is *observed*; otherwise it is *hidden*



- **Directed edge** representing dependency among variables:
  - A Directed Acyclic Graphical (DAG) model
- Using plate notation instead of flat notation
  - The variables in the same plate share the same **conditional probability table**

Flat notation

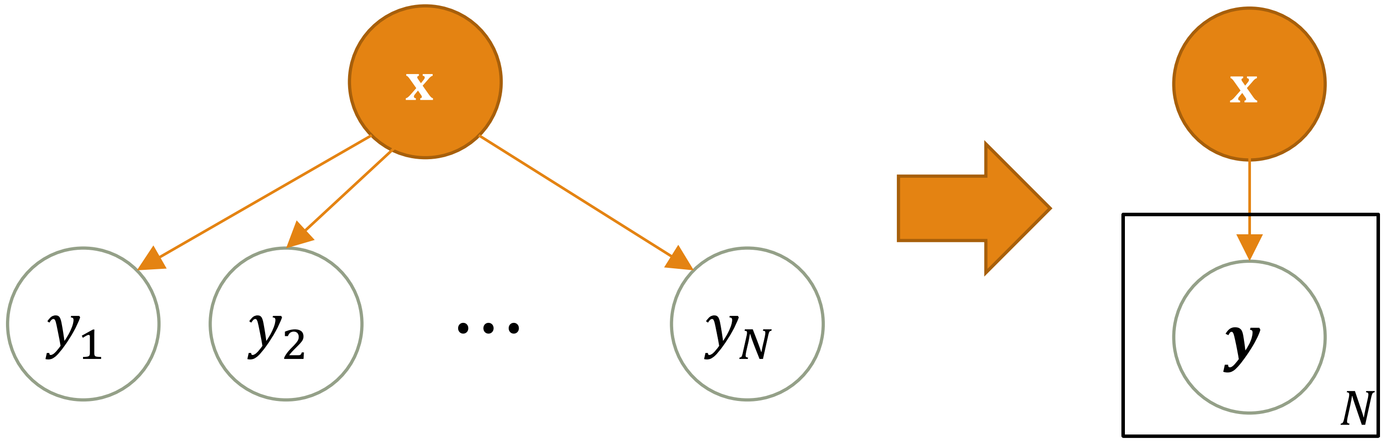


Plate notation

An Example of Plate Notation

Flat notation

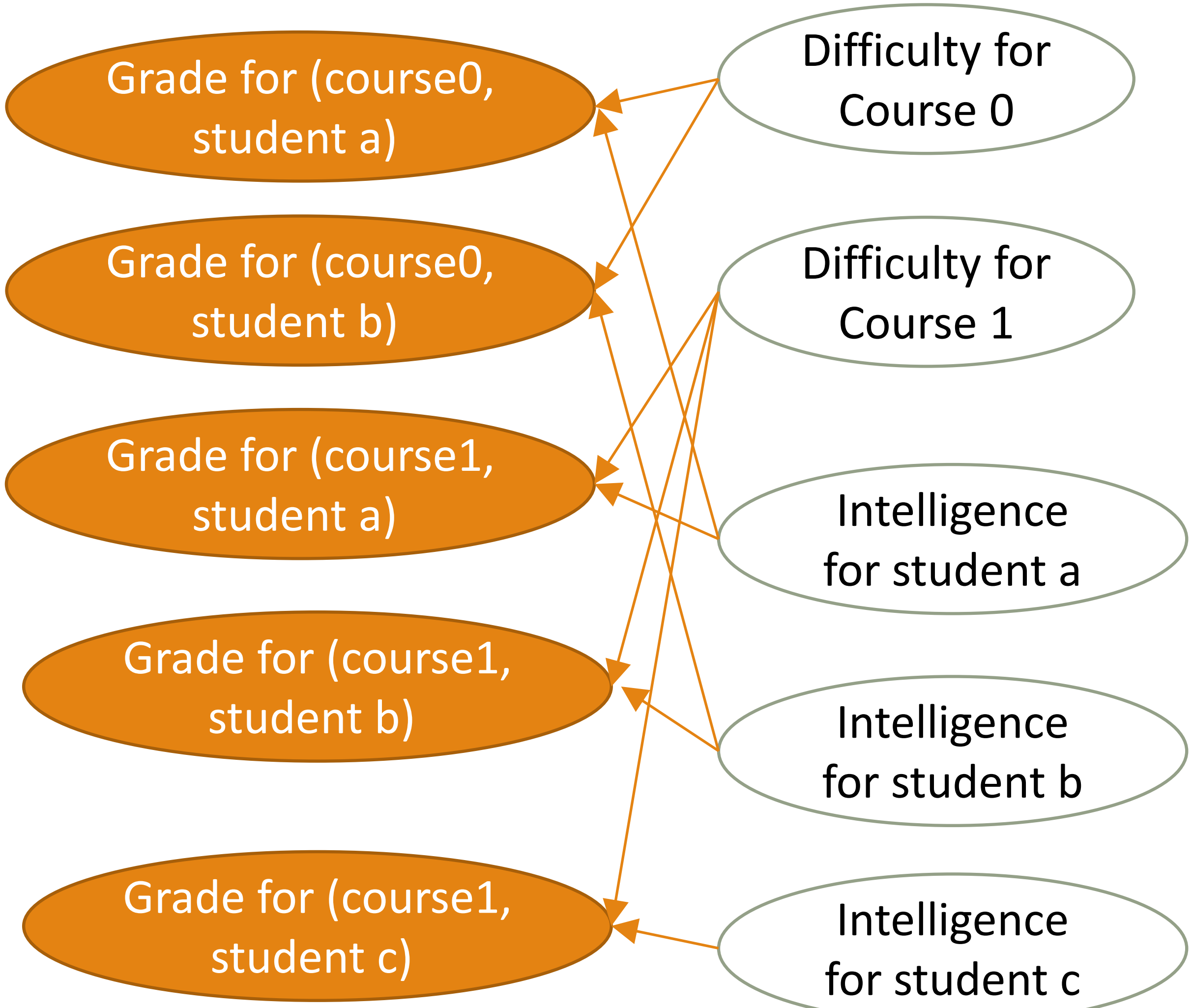
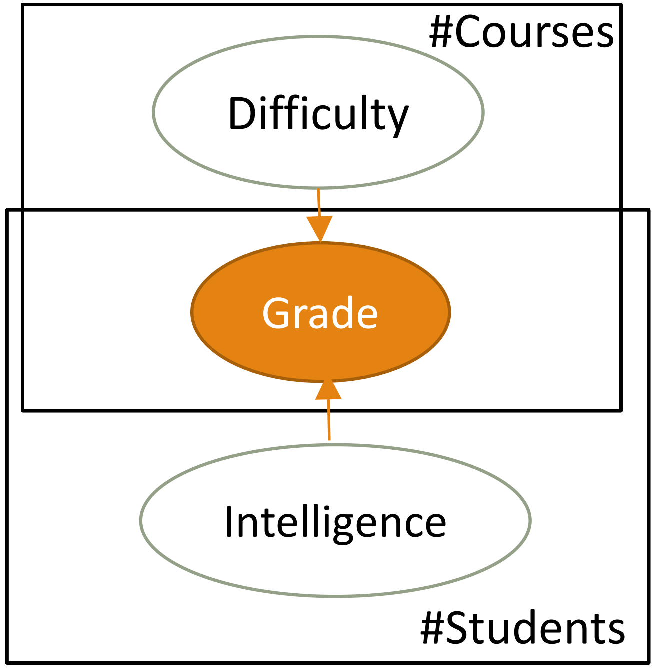


Plate notation



Summary

Summary

- Bayes Classifier
- Bayesian Networks

Recommended Readings

- Heckerman, D. (1999). A tutorial on learning with Bayesian networks. *Learning in Graphical Models, M. Jordan ed.* Retrieved from https://www.microsoft.com/en-us/research/publication/a-tutorial-on-learning-with-bayesian-networks/?from=http%3A%2F%2Fresearch.microsoft.com%2Fen-us%2Fum%2Fpeople%2Fheckerman%2Ftutorial.pdf
- Lewis, D. D. (1998). Naive (Bayes) at forty: The independence assumption in information retrieval. *Machine Learning, ECML-98, 1398*, 4-15. Retrieved from https://link.springer.com/chapter/10.1007/BFb0026666#citeas
- Lim, T.-S., Loh, W.-Y., & Shih, Y.-S. (2000). A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms. *Machine Learning, 40*(3), 203–228. retrieved from https://link.springer.com/article/10.1023/A:1007608224229
- Murphy, K. P. (2012). *Machine learning: A probabilistic perspective*. Cambridge, MA: The MIT Press.
- Ng, A. Y. & Jordan, M. I. (2002). On discriminative vs. generative classifiers: A comparison of logistic regression and naïve Bayes. *NIPS.* Retrieved from https://papers.nips.cc/paper/2020-on-discriminative-vs-generative-classifiers-a-comparison-of-logistic-regression-and-naive-bayes
- Pearl, J. (2000). *Causality: Models, reasoning, and inference.* Cambridge, UK: Cambridge Univ. Press.
- Russell, S. & Norvig, P. (2003). *Artificial intelligence: A modern approach (2nd ed.).* New York, NY: Prentice Hall.

References

- All other multimedia elements belong to © 2018 University of Illinois Board of Trustees.

Appendix: Detailed Computation of Some Probabilities on Slide #16 “Bayesian Network: An Example”

$$
\begin{aligned}
s(F=1|S=1) &= \frac{p(S=1,F=1)}{p(S=1)} \\
&= \frac{p(S=1|F=1)\times p(F=1)}{\sum_f p(S=1,F)} \\
&= \frac{p(S=1|F=1)\times p(F=1)}{p(S=1|F=1)p(F=1)+p(S=1|F=0)p(F=0)} \\
&= \frac{0.9\times 0.02}{0.9\times 0.02+0.01\times 0.98} \\
&= 0.647
\end{aligned}
$$

$$
\begin{aligned}
p(T=1|F=1,A=1) &= \frac{p(T=1,F=1,A=1)}{p(A=1|F=1)p(F=1)} \\
&= \frac{p(A=1|F=1,T=1)p(F=1)p(T=1)}{p(A=1|F=1)p(F=1)} \\
&= \frac{p(A=1|F=1,T=1)p(T=1)}{\sum_T p(A=1,T|F=1)} \\
&= \frac{0.5\times 0.1}{0.5\times 0.1+0.99\times 0.9} \\
&= 0.053
\end{aligned}
$$

$$
\begin{aligned}
p(T=1|F=0,A=1) &= \frac{p(A=1|T=1,F=0)p(T=1)}{\sum_T p(A=1,T|F=0)} \\
&= \frac{0.85\times 0.1}{0.85\times 0.1+0.001\times 0.9} \\
&= 0.9989 \sim 1.000
\end{aligned}
$$