CS249: ADVANCED DATA MINING

Probabilistic Classifiers and Naïve Bayes

Instructor: Yizhou Sun

yzsun@cs.ucla.edu

April 24, 2017

Announcements

- Homework 1
  - Due end of the day of this Friday (11:59pm)

- Reminder of late submission policy
  - original score * $\mathbf{1}(t <= 24)e^{-(ln(2)/12)*t}$
  - E.g., if you are t = 12 hours late, maximum of half score will be obtained; if you are 24 hours late, 0 score will be given.

Methods to Learn: Last Lecture

	Vector Data	Text Data	Recommender System	Graph & Network
Classification	**Decision Tree**; Naïve Bayes; **Logistic Regression** SVM; NN			Label Propagation
Clustering	K-means; hierarchical clustering; DBSCAN; Mixture Models; kernel k-means	PLSA; LDA	Matrix Factorization	SCAN; Spectral Clustering
Prediction	**Linear Regression GLM**		Collaborative Filtering	
Ranking				PageRank
Feature Representation		Word embedding		Network embedding

Methods to Learn

	Vector Data	Text Data	Recommender System	Graph & Network
Classification	**Decision Tree**; **Naïve Bayes**; **Logistic Regression** SVM; NN			Label Propagation
Clustering	K-means; hierarchical clustering; DBSCAN; Mixture Models; kernel k-means	PLSA; LDA	Matrix Factorization	SCAN; Spectral Clustering
Prediction	**Linear Regression** **GLM**		Collaborative Filtering	
Ranking				PageRank
Feature Representation		Word embedding		Network embedding

Probabilistic Classifiers and Naïve Bayes

- Probabilistic Classifiers
- Naïve Bayes
- Bayesian Network
- Summary

Basic Probability Review

- Have two dice h_1 and h_2
- The probability of rolling an *i* given die h_1 is denoted $P(i|h_1)$. This is a *conditional probability*
- Pick a die at random with probability $P(h_j)$, j=1 or 2. The probability for picking die h_j and rolling an i with it is called *joint probability* and is $P(i, h_j)=P(h_j)P(i|h_j)$.
- If we know $P(i|h_j)$, then the so-called *marginal probability* $P(i)$ can be computed as: $P(i) = \sum_j P(i, h_j)$
- For any X and Y, $P(X,Y)=P(X|Y)P(Y)$

Bayes' Theorem: Basics

- Bayes' Theorem: $P(h|\mathbf{X}) = \frac{P(\mathbf{X}|h)P(h)}{P(\mathbf{X})}$
  - Let **X** be a data sample ("*evidence*")
  - Let h be a *hypothesis* that X belongs to class C
  - P(h) (*prior probability*): the initial probability
    - E.g., **X** will buy computer, regardless of age, income, …
  - 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
  - P(**X**): marginal probability that sample data is observed
    - $P(X) = \sum_h P(X|h)\, P(h)$
  - P(h|**X**), (i.e., *posterior probability*): the probability that the hypothesis holds given the observed data sample **X**

Classification: Choosing Hypotheses

- *Maximum Likelihood* (maximize the likelihood):

$$h_{ML} = \arg\max_{h \in H} P(X \mid h)$$

- *Maximum a posteriori* (maximize the posterior):
  - Useful observation: it does not depend on the denominator P(X)

$$h_{MAP} = \arg\max_{h \in H} P(h \mid X) = \arg\max_{h \in H} P(X \mid h) P(h)$$

Classification by Maximum A Posteriori

- Let D be a training set of tuples and their associated class labels, and each tuple is represented by an p-D attribute vector $\mathbf{X} = (x_1, x_2, \ldots, x_p)$
- Suppose there are *m* classes $Y \in \{C_1, C_2, \ldots, C_m\}$
- Classification is to derive the maximum posteriori, i.e., the maximal $P(Y=C_j|\mathbf{X})$
- This can be derived from Bayes' theorem $P(Y=C_j|\mathbf{X}) = \frac{P(\mathbf{X}|Y=C_j)P(Y=C_j)}{P(\mathbf{X})}$
- Since P(X) is constant for all classes, only $P(y,\mathbf{X}) = P(\mathbf{X}|y)P(y)$ needs to be maximized

Example: Cancer Diagnosis

- A patient takes a lab test and the result comes back positive. It is known that
  - a correct positive result in only 98% of the cases
    - $P(test = + | cancer) = .98$
  - a correct negative result in only 97% of the cases
    - $P(test = - | \neg cancer) = .97$
  - only 0.008 of the entire population has this disease
    - $P(cancer) = .008$

1. What is the probability that this patient has cancer?
2. What is the probability that he does not have cancer?
3. What is the diagnosis?

Solution

$P(cancer) = .008$	$P(\neg cancer) = .992$
$P(test = + \mid cancer) = .98$	$P(test = - \mid cancer) = .02$
$P(test = + \mid \neg cancer) = .03$	$P(test = - \mid \neg cancer) = .97$

How do we know these parameters in practice?

Using Bayes Formula:

$P(cancer \mid test = +) = P(test = + \mid cancer) x P(cancer) / P(test = +)$

$= 0.98 \times 0.008 / P(test = +) = .00784 / P(test = +)$

$P(\neg cancer \mid test = +) = P(test = + \mid \neg cancer) x P(\neg cancer) / P(test = +)$

$= 0.03 \times 0.992 / P(test = +) = .0298 / P(test = +)$

So, the patient most likely does not have cancer.

Probabilistic Classifiers and Naïve Bayes

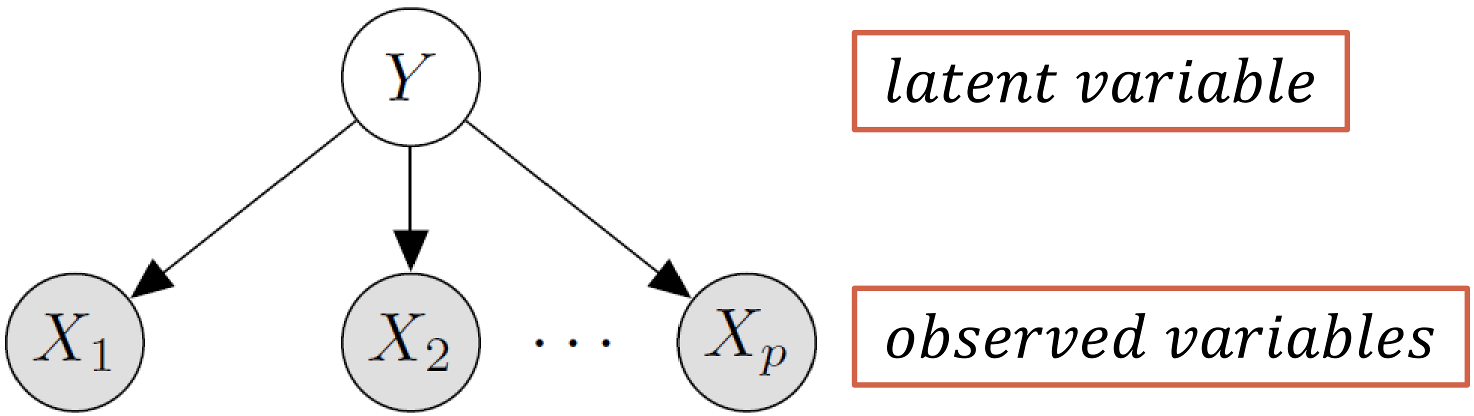
- Probabilistic Classifiers
- Naïve Bayes
- Bayesian Network
- Summary

Naïve Bayes Classifier

- Let D be a training set of tuples and their associated class labels, and each tuple is represented by an p-D attribute vector **X** = (x_1, x_2, …, x_p)
- Suppose there are *m* classes Y∈{C_1, C_2, …, C_m}
- Goal: Find Y
  $\max_Y P(Y|\boldsymbol{X}) = P(Y, \boldsymbol{X})/P(\boldsymbol{X}) \propto P(\boldsymbol{X}|Y)P(Y)$
- A simplified assumption: attributes are conditionally independent given the class (class conditional independency):
  - $P(\boldsymbol{X}|Y) = \prod_k P(x_k|Y)$

Conditional independence Assumption

- Graphical model illustration



Estimate Parameters by MLE

- Given a dataset $D = \{(\boldsymbol{X}^{(i)}, Y^{(i)})\}$, the goal is to
  - Find the best estimators $P(C_j)$ and $P(X_k = x_k|C_j)$, for every $j = 1, \ldots, m$ *and* $k = 1, \ldots, p$
  - that maximizes the likelihood of observing D:

$$L = \prod_i P\left(\boldsymbol{X}^{(i)}, Y^{(i)}\right) = \prod_i P\left(\boldsymbol{X}^{(i)}|Y^{(i)}\right) P(Y^{(i)})$$

$$= \prod_i (\prod_k P\left(X_k^{(i)}|Y^{(i)}\right)) P(Y^{(i)})$$

- Estimators of Parameters:
  - $P(C_j) = |C_{j,D}|/|D|$ ($|C_{j,D}|$= # of tuples of C_j in D) (why?)
  - $P(X_k = x_k|C_j)$: X_k can be either discrete or numerical

Discrete and Continuous Attributes

- If X_k is discrete, with V possible values
  - $P(x_k | C_j)$ is the # of tuples in C_j having value x_k for X_k divided by $|C_{j, D}|$
- If X_k is continuous, with observations of real values
  - $P(x_k | C_j)$ is usually computed based on Gaussian distribution with a mean μ and standard deviation σ
  - Estimate (μ, σ^2) according to the observed X in the category of C_j
    - Sample mean and sample variance
  - $P(x_k | C_j)$ is then $P(X_k = x_k | C_j) = f(x_k | \mu_{C_j}, \sigma_{C_j})$

Gaussian density function

Naïve Bayes Classifier: Training Dataset

Class:

C1:buys_xbox = 'yes'

C2:buys_xbox = 'no'

Data to be classified:

X = (age <=30,

Income = medium,

Student = yes

Credit_rating = Fair)

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

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

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

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

Avoiding the Zero-Probability Problem

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

$$P(X|C_j) = \prod_{k=1}^{p} P(x_k|C_j)$$

- Use **Laplacian correction** (or Laplacian smoothing)
  - *Adding 1 to each case*
    - $P(x_k = v|C_j) = \frac{n_{jk,v}+1}{|C_{j,D}|+V}$ where $n_{jk,v}$ is # of tuples in C_j having value $x_k = \mathrm{v}$, V is the total number of values that can be taken
    - Ex. Suppose a training dataset with 1000 tuples, for category "buys_xbox = yes", income=low (0), income= medium (990), and income = high (10)

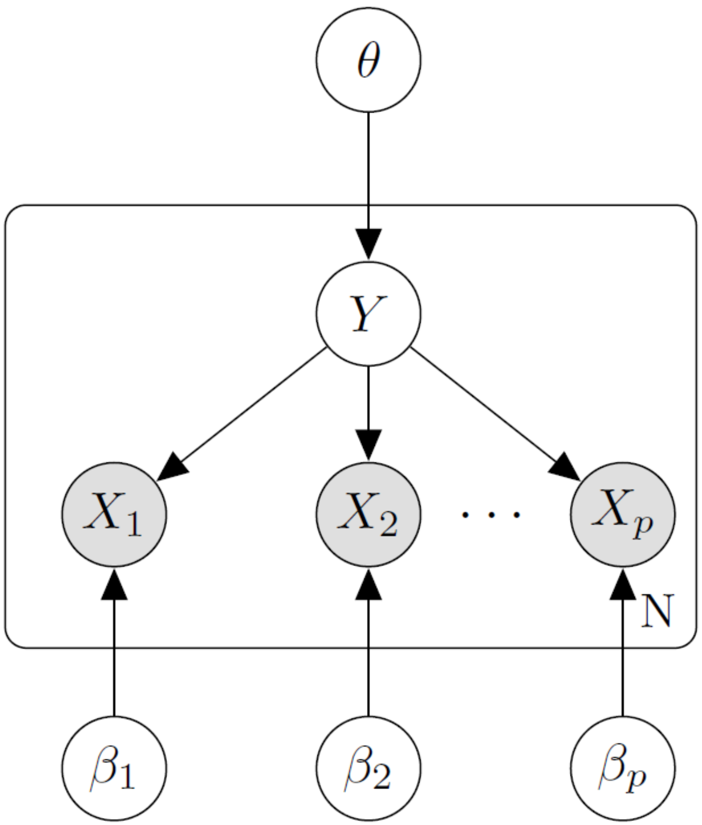
      Prob(income = low|buys_xbox = "yes") = 1/1003

      Prob(income = medium|buys_xbox = "yes") = 991/1003

      Prob(income = high|buys_xbox = "yes") = 11/1003

  - The "corrected" prob. estimates are close to their "uncorrected" counterparts

A Generative Model View



- For each data point
  - Draw $Y \sim Discrete(\theta),\ i.e., P(Y = C_j) = \theta_j$
  - For each attribute X_k
    - Draw $X_k \sim p(X_k | \beta_k, Y)$

- Likelihood
- $L = \prod_i p(x^{(i)}, y^{(i)} | \theta, \beta)$
  $= \prod_i p(x^{(i)} | y^{(i)}, \beta) p(y^{(i)} | \theta)$
  $= \prod_i \prod_k p\left(x_k^{(i)} \middle| y^{(i)}, \beta\right) p(y^{(i)} | \theta)$

Smoothing and Prior on Attribute Distribution

- $Discrete\ distribution$: $X_k|Y = C_j \sim \boldsymbol{\beta}$ (short for $\boldsymbol{\beta}_k^j$)
  - $P\left(X_k = v \middle| C_j, \boldsymbol{\beta}\right) = \boldsymbol{\beta}_v$
- Put prior to $\boldsymbol{\beta}$
  - In discrete case, the prior can be chosen as symmetric Dirichlet distribution: $\boldsymbol{\beta} \sim Dir(\alpha),\ i.e., P(\boldsymbol{\beta}) \propto \prod_v \boldsymbol{\beta}_v^{\alpha-1}$
  - $posterior\ distribution$:
    - $P\left(\boldsymbol{\beta} \middle| X_{1k}, \dots, X_{nk}, Y = C_j\right) \propto P\left(X_{1k}, \dots, X_{nk} \middle| C_j, \boldsymbol{\beta}\right) P(\boldsymbol{\beta})$, another Dirichlet distribution, with new parameter $(\alpha + c_1, \dots, \alpha + c_v, \dots, \alpha + c_V)$
    - c_v is the number of observations taking value v
  - Inference: $P\left(X_k = v \middle| X_{1k}, \dots, X_{nk}, C_j\right) = \int P(X_k = v|\boldsymbol{\beta}) P\left(\boldsymbol{\beta} \middle| X_{1k}, \dots, X_{nk}, C_j\right) \mathrm{d}\boldsymbol{\beta} = \frac{\boldsymbol{c_v + \alpha}}{\boldsymbol{\sum c_v + V\alpha}}$
    - Equivalent to adding α to each observation value v

Notes on Parameter Learning

- Why the probability of $P\left(X_k \middle| C_j\right)$ is estimated in this way?
  - http://www.cs.columbia.edu/~mcollins/em.pdf
  - http://www.cs.ubc.ca/~murphyk/Teaching/CS340-Fall06/reading/NB.pdf

Naïve Bayes Classifier: Comments

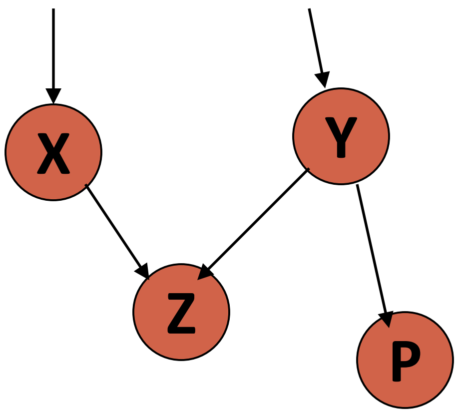
- Advantages
  - Easy to implement
  - Good results obtained in most of the cases
- Disadvantages
  - Assumption: class conditional independence, therefore loss of accuracy
  - Practically, dependencies exist among variables
    - E.g., Patients profile: age, family history, etc.; Symptoms: fever, cough 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

Probabilistic Classifiers and Naïve Bayes

- Probabilistic Classifiers
- Naïve Bayes
- Bayesian Network
- Summary

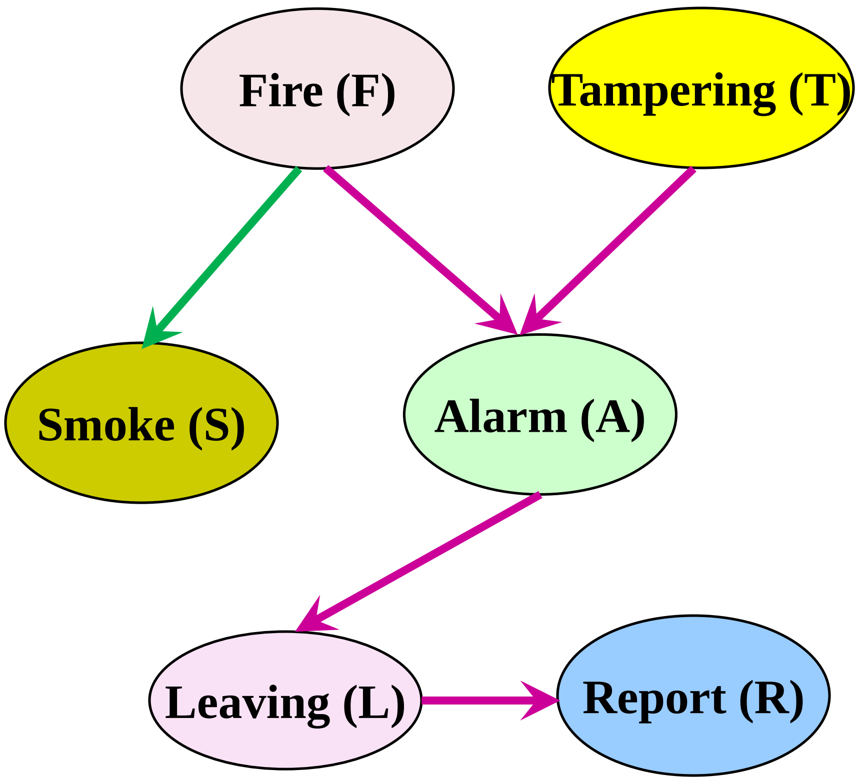
Bayesian Belief Networks (BNs)

- **Bayesian belief network** (also known as **Bayesian network**, **probabilistic network**): allows *class conditional independencies* between *subsets* of variables
- Two components: (1) A *directed acyclic graph* (called a structure) and (2) a set of *conditional probability tables* (CPTs)
- A (*directed acyclic*) graphical model of *causal influence* relationships
  - Represents dependency among the variables
  - Gives a specification of joint probability distribution



- ❑ Nodes: random variables
- ❑ Links: dependency
- ❑ X and Y are the parents of Z, and Y is the parent of P
- ❑ No dependency between Z and P conditional on Y
- ❑ Has no cycles

A Bayesian Network and Some of Its CPTs



CPT: Conditional Probability Tables

	F	¬F
S	.90	.01
¬S	.10	.99

	F, T	F, ¬T	¬F, T	¬F, ¬T
A	.5	.99	.85	.0001
¬A	.95	.01	.15	.9999

CPT shows the conditional probability for each possible combination of its parents

Derivation of the probability of a particular combination of values of **X**, from CPT (joint probability):

$$P(x_1,\ldots,x_n) = \prod_{i=1}^{n} P(x_i \mid Parents(x_i))$$

*Inference in Bayesian Networks

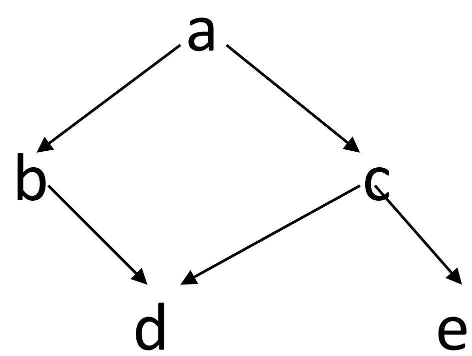
- Infer the probability of values of some variable given the observations of other variables
  - E.g., P(Fire = True|Report = True, Smoke = True)?
- Computation
  - Exact computation by enumeration
  - In general, the problem is NP hard
    - *Approximation algorithms are needed

*Inference by enumeration

- To compute posterior marginal $P(X_i \mid E=e)$
  - Add all of the terms (atomic event probabilities) from the full joint distribution
  - If **E** are the evidence (observed) variables and **Y** are the other (unobserved) variables, then:

    $P(X|\mathbf{e}) = \alpha\, P(X, \mathbf{E}) = \alpha \sum P(X, \mathbf{E}, \mathbf{Y})$
  - Each P(X, **E**, **Y**) term can be computed using the chain rule
- Computationally expensive!

*Example: Enumeration



- $P(d|e) = \alpha\, \Sigma_{ABC} P(a, b, c, d, e)$
  $= \alpha\, \Sigma_{ABC} P(a)\, P(b|a)\, P(c|a)\, P(d|b,c)\, P(e|c)$
- With simple iteration to compute this expression, there's going to be a lot of repetition (e.g., P(e|c) has to be recomputed every time we iterate over C=true)
  - *A solution: variable elimination

*How Are Bayesian Networks Constructed?

- **Subjective construction**: 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
  - Markovian assumption: Each variable becomes independent of its non-effects once its direct causes are known
  - E.g., S ← F → A ← T, path S→A is blocked once we know F→A
- **Synthesis from other specifications**
  - E.g., from a formal system design: block diagrams & info flow
- **Learning from data**
  - E.g., from medical records or student admission record
  - Learn parameters give its structure or learn both structure and parms
  - Maximum likelihood principle: favors Bayesian networks that maximize the probability of observing the given data set

*Learning Bayesian Networks: Several Scenarios

- Scenario 1: Given both the network structure and all variables observable: *compute only the CPT entries (Easiest case!)*
- Scenario 2: Network structure known, some variables hidden: *gradient descent* (greedy hill-climbing) method, i.e., search for a solution along the steepest descent of a criterion function
  - Weights are initialized to random probability values
  - At each iteration, it moves towards what appears to be the best solution at the moment, w.o. backtracking
  - Weights are updated at each iteration & converge to local optimum
- 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 Classifiers and Naïve Bayes

- Probabilistic Classifiers
- Naïve Bayes
- Bayesian Network
- Summary

Summary

- Probabilistic Classifiers
  - Classification ⇔ hypothesis selection in probabilistic models
- Naïve Bayes
  - Conditional independence assumption
  - MLE for parameters
  - Laplace smooth
- Bayesian Networks
  - Joint probability computation; CPT

Course Project

- Team Sign-up (Participation)
- Proposal (5%)
- Presentation (15%, in class peer review)
- Final Report (15%)

Proposal

- What to submit: A 2-Page proposal including
- 1. Problem and goal
  - What do you want to solve?
  - Why do you think it is important?
  - What results do you expect?
- 2. Formalization into data mining task
  - Which data type?
  - Which function? E.g., Frequent pattern mining, classification, and clustering.
- 3. Data plan
  - What kind of data?
  - Where and how do you get the data?
  - Make sure get data in time
- 4. Schedule: detailed plan of your project

Collaboration Rules

- Every member in a team gets the same score (encourage teamwork)
  - Exception: the team has the right to claim someone as a free rider, and we will lower his/her score
- A table describing your workload distribution

Task	People
1. Collecting and preprocessing data	Student A
2. Implementing Algorithm 1	Student B
3. Implementing Algorithm 2	Student C and D
4. Evaluating and comparing algorithms	Student A
5. Writing report	Student B and C
6. Slides, demo, and Presentation	student A, B

- Peer Evaluation

Past Projects

- Outlier Detection from Clinical Lab Data
- COURSE PLANNER
- Stylometry Classification for Authors
- Price Range Prediction for Real Estate Data
- Student Application Recommendation System
- ……

Datasets

- UCI Machine Learning Repository
  - http://archive.ics.uci.edu/ml/
- Bibliographic data
  - https://aminer.org/citation
- Wikipedia
  - https://figshare.com/articles/Wikipedia_Clickstream/1305770
  - https://dumps.wikimedia.org/

Project Ideas

- Wikipedia
  - Page classification: Person vs not a person
  - Hyperlink prediction
- Are there discriminations in current data mining algorithms
  - E.g., decision boundary is unfair for student admission case
  - E.g., different words are picked for describing the same concept for different gender