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' Rule

$$p(h \mid d) = \frac{P(d \mid h)P(h)}{P(d)}$$

Understanding Bayes' rule

d = data

h = hypothesis (model)

- rearranging

$p(h \mid d)P(d) = P(d \mid h)P(h)$

$P(d,h) = P(d,h)$

the same joint probability

on both sides

LEARN

Who is who in Bayes' rule

$P(h)$: prior belief (probability of hypothesis *h* before seeing any data)

$P(d \mid h)$: likelihood (probability of the data if the hypothesis *h* is true)

$P(d) = \sum_h P(d \mid h)P(h)$: data evidence (marginal probability of the data)

$P(h \mid d)$: posterior (probability of hypothesis *h* after having seen the data *d*)

Example of Bayes Theorem

- Given:
  - A doctor knows that meningitis causes stiff neck 50% of the time
  - Prior probability of any patient having meningitis is 1/50,000
  - Prior probability of any patient having stiff neck is 1/20

- If a patient has stiff neck, what's the probability he/she has meningitis?

$$P(M \mid S) = \frac{P(S \mid M)P(M)}{P(S)} = \frac{0.5 \times 1/50000}{1/20} = 0.0002$$

Choosing Hypotheses

- *Maximum Likelihood* hypothesis:

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

- Generally we want the most probable hypothesis given training data. This is the *maximum a posteriori* hypothesis:
  - Useful observation: it does not depend on the denominator P(d)

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

Bayesian Classifiers

- Consider each attribute and class label as random variables

- Given a record with attributes (A_1, A_2,…,A_n)
  - Goal is to predict class C
  - Specifically, we want to find the value of C that maximizes $P(C| A_1, A_2,…,A_n)$

- Can we estimate $P(C| A_1, A_2,…,A_n)$ directly from data?

Bayesian Classifiers

- Approach:
  - compute the posterior probability $P(C \mid A_1, A_2, \ldots, A_n)$ for all values of C using the Bayes theorem

$$\boxed{P(C \mid A_1 A_2 \ldots A_n) = \frac{P(A_1 A_2 \ldots A_n \mid C) P(C)}{P(A_1 A_2 \ldots A_n)}}$$

  - Choose value of C that maximizes
    $P(C \mid A_1, A_2, \ldots, A_n)$
  - Equivalent to choosing value of C that maximizes
    $P(A_1, A_2, \ldots, A_n \mid C)\, P(C)$
- How to estimate $P(A_1, A_2, \ldots, A_n \mid C)$?

Naïve Bayes Classifier

- Assume independence among attributes A_i when class is given:
  - $P(A_1, A_2, \ldots, A_n | C) = P(A_1 | C_j)\, P(A_2 | C_j) \ldots P(A_n | C_j)$
  - Can estimate $P(A_i | C_j)$ for all A_i and C_j.
  - This is a simplifying assumption which may be violated in reality
- The Bayesian classifier that uses the Naïve Bayes assumption and computes the MAP hypothesis is called Naïve Bayes classifier

$$\boxed{c_{Naive\ Bayes} = \arg\max_{c} P(c)P(\mathbf{x} | c) = \arg\max_{c} P(c)\prod_{i} P(a_i | c)}$$

How to Estimate Probabilities from Data?

Tid	Refund	Marital Status	Taxable Income	Evade
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

- Class: $P(C) = N_c/N$
  - e.g., P(No) = 7/10,
    P(Yes) = 3/10
- For discrete attributes:

$$P(A_i \mid C_k) = |A_{ik}| / N_{c_k}$$

  - where $|A_{ik}|$ is number of instances having attribute A_i and belongs to class C_k
  - Examples:

P(Status=Married|No) = 4/7
P(Refund=Yes|Yes)=0

How to Estimate Probabilities from Data?

- For continuous attributes:
  - Discretize the range into bins
    - one ordinal attribute per bin
    - violates independence assumption
  - Two-way split: $(A < v)$ or $(A > v)$
    - choose only one of the two splits as new attribute
  - Probability density estimation:
    - Assume attribute follows a normal distribution
    - Use data to estimate parameters of distribution (e.g., mean and standard deviation)
    - Once probability distribution is known, can use it to estimate the conditional probability $P(A_i|c)$

How to Estimate Probabilities from Data?

Tid	Refund	Marital Status	Taxable Income	Evade
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

- Normal distribution:

$$P(A_i \mid c_j) = \frac{1}{\sqrt{2\pi\sigma_{ij}^2}} e^{-\frac{(A_i-\mu_{ij})^2}{2\sigma_{ij}^2}}$$

  - One for each (A_i,c_i) pair

- For (Income, Class=No):
  - If Class=No
    - sample mean = 110
    - sample variance = 2975

$$P(Income = 120 \mid No) = \frac{1}{\sqrt{2\pi}(54.54)} e^{-\frac{(120-110)^2}{2(2975)}} = 0.0072$$

Naïve Bayesian Classifier: Training Dataset

Binary Classificat

Class:

C1:buys_computer = 'yes'

C2:buys_computer = 'no'

New Data:

X = (age <=30,

Income = medium,

Student = yes

Credit_rating = Fair)

Query: X

age (x_1)	income (x_2)	student (x_3)	credit_rating (x_4)	comp (y_1)
<=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 Bayesian Classifier: An Example

Buy OR NOT BUY

Query

Given X (age=youth, income=medium, student=yes, credit=fair)

Maximize P(X|Ci)P(Ci), for i=1,2

First step: Compute P(C) The prior probability of each class can be computed based on the training tuples:

P(buys_computer=yes)=9/14=0.643 $P(C_1)$

P(buys_computer=no)=5/14=0.357 $P(C_2)$

$$P(C_K|X) \propto \left\{ P(C_K) \prod_{i=1}^{4} P(X_i|C_K) \right\}$$

max

$K \in \{1,2\}$

$C = \{\overset{1}{Yes}, \overset{2}{No}\}$.

Naïve Bayesian Classifier: An Example

for(k ← 1 → 2){

C_1 yes

Given X (age=youth, income=medium, student=yes, credit=fair)

Maximize P(X|Ci)P(Ci), for i=1,2

Second step: compute P(X|Ci)

P(**X|buys_computer=yes**)= P(age=youth|buys_computer=yes)x
P(income=medium|buys_computer=yes) x
P(student=yes|buys_computer=yes)x
P(credit_rating=fair|buys_computer=yes)
= 0.044

P(age=youth|buys_computer=yes)=0.222
P(income=medium|buys_computer=yes)=0.444
P(student=yes|buys_computer=yes)=6/9=0.667
P(credit_rating=fair|buys_computer=yes)=6/9=0.667

Naïve Bayesian Classifier: An Example

Given X (age=youth, income=medium, student=yes, credit=fair)

Maximize P(X|Ci)P(Ci), for i=1,2

Second step: compute P(X|Ci)

P(**X|buys_computer=no**) = P(age=youth|buys_computer=no)x
P(income=medium|buys_computer=no) x
P(student=yes|buys_computer=no) x
P(credit_rating=fair|buys_computer=no)
= 0.019

P(age=youth|buys_computer=no)=3/5=0.666

P(income=medium|buys_computer=no)=2/5=0.400

P(student=yes|buys_computer=no)=1/5=0.200

P(credit_rating=fair|buys_computer=no)=2/5=0.400

Naïve Bayesian Classifier: An Example

Given X (age=youth, income=medium, student=yes, credit=fair)

Maximize P(X|Ci)P(Ci), for i=1,2

We have computed in the first and second steps:

P(buys_computer=yes)=9/14=0.643

P(buys_computer=no)=5/14=0.357

P(X|buys_computer=yes)= 0.044

P(X|buys_computer=no)= 0.019

Third step: compute **P(X|Ci)P(Ci)** for each class

P(X|buys_computer=yes)P(buys_computer=yes)=0.044 x 0.643=0.028

P(X|buys_computer=no)P(buys_computer=no)=0.019 x 0.357=0.007

The naïve Bayesian Classifier predicts **X belongs to class ("buys_computer = yes")**

0.028 > 0.007

Example

x_1 x_2 x_3 c

Tid	Refund	Marital Status	Taxable Income	Evade
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Training set :
(Öğrenme Kümesi)

$c = \{Yes, No\}$

Binary Classification

k

Given a Test Record:

$$X = (\text{Refund} = \text{No}, \text{Married}, \text{Income} = 120\text{K})$$

Query

Example of Naïve Bayes Classifier

Given a Test Record:

$$X = (\text{Refund} = \text{No}, \text{Married}, \text{Income} = 120\text{K})$$

Ans-NO

naive Bayes Classifier:

P(Refund=Yes|No) = 3/7
P(Refund=No|No) = 4/7
P(Refund=Yes|Yes) = 0
P(Refund=No|Yes) = 1
P(Marital Status=Single|No) = 2/7
P(Marital Status=Divorced|No)=1/7
P(Marital Status=Married|No) = 4/7
P(Marital Status=Single|Yes) = 2/7
P(Marital Status=Divorced|Yes)=1/7
P(Marital Status=Married|Yes) = 0

For taxable income:
If class=No: sample mean=110
sample variance=2975
If class=Yes: sample mean=90
sample variance=25

- P(X|Class=No) = P(Refund=No|Class=No)
  $\times$ P(Married| Class=No)
  $\times$ P(Income=120K| Class=No)
  = 4/7 $\times$ 4/7 $\times$ 0.0072 = 0.0024

- P(X|Class=Yes) = P(Refund=No| Class=Yes)
  $\times$ P(Married| Class=Yes)
  $\times$ P(Income=120K| Class=Yes)
  = 1 $\times$ 0 $\times$ 1.2 $\times$ 10^{-9} = 0

Since P(X|No)P(No) > P(X|Yes)P(Yes)

Therefore P(No|X) > P(Yes|X)

=> Class = No

Avoiding the 0-Probability Problem

- If one of the conditional probability is zero, then the entire expression becomes zero
- Probability estimation:

$$\text{Original}: P(A_i \mid C) = \frac{N_{ic}}{N_c}$$

$$\text{Laplace}: P(A_i \mid C) = \frac{N_{ic} + 1}{N_c + c}$$

$$\text{m - estimate}: P(A_i \mid C) = \frac{N_{ic} + mp}{N_c + m}$$

c: number of classes

p: prior probability

m: parameter

Naïve Bayes (Summary)

- *Advantage*
  - Robust to isolated noise points
  - Handle missing values by ignoring the instance during probability estimate calculations
  - Robust to irrelevant attributes
- *Disadvantage*
  - Assumption: class conditional independence, which may cause loss of accuracy
  - Independence assumption may not hold for some attribute. Practically, dependencies exist among variables
    - Use other techniques such as Bayesian Belief Networks (BBN)

Remember

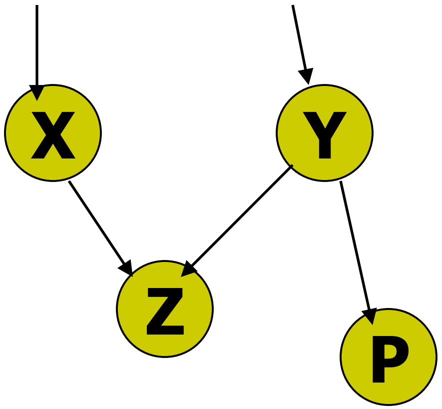
- Bayes' rule can be turned into a classifier
- Maximum A Posteriori (MAP) hypothesis estimation incorporates prior knowledge; Max Likelihood (ML) doesn't
- Naive Bayes Classifier is a simple but effective Bayesian classifier for vector data (i.e. data with several attributes) that assumes that attributes are independent given the class.
- Bayesian classification is a generative approach to classification

Classification Paradigms

- In fact, we can categorize three fundamental approaches to classification:
- Generative models: Model $p(x|C_k)$ and $P(C_k)$ separately and use the Bayes theorem to find the posterior probabilities $P(C_k|x)$
  - E.g. Naive Bayes, Gaussian Mixture Models, Hidden Markov Models,…
- Discriminative models:
  - Determine $P(C_k|x)$ directly and use in decision
  - E.g. Linear discriminant analysis, SVMs, NNs,…
- Find a discriminant function f that maps x onto a class label directly without calculating probabilities

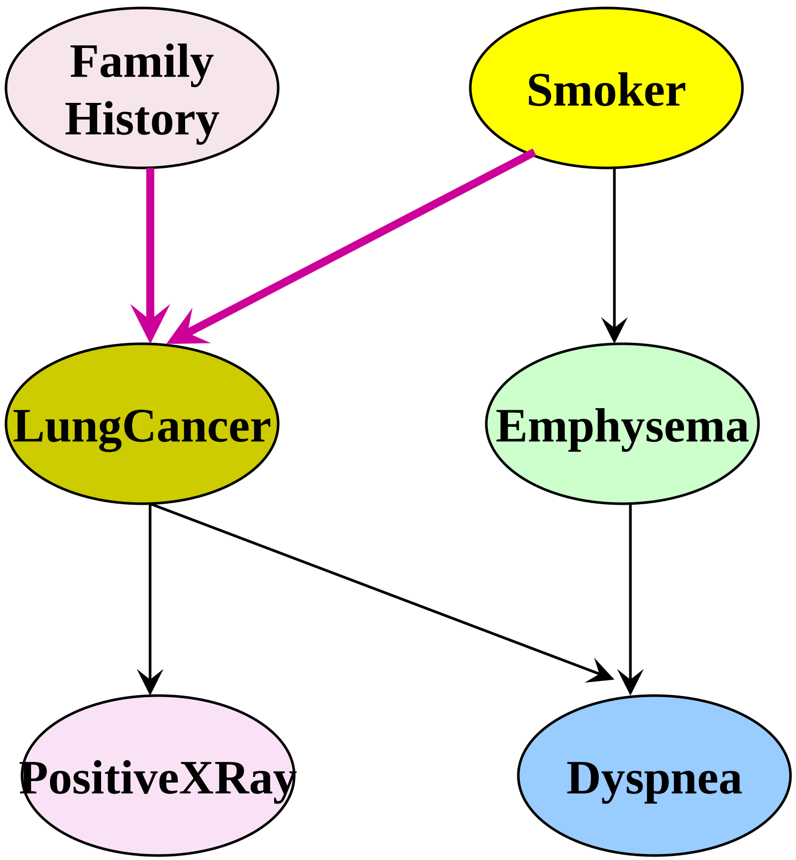
Bayesian Belief Networks

- Bayesian belief network allows a *subset* of the variables to be conditionally independent
- A graphical model of causal relationships *(neden sonuç ilişkilerini simgeleyen bir çizge tabanlı model)*
  - Represents <u>dependency</u> 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
- ❑ Has no loops or cycles

Bayesian Belief Network: An Example



Bayesian Belief Networks

The **conditional probability table** (**CPT**) for variable LungCancer:

	(FH, S)	(FH, ~S)	(~FH, S)	(~FH, ~S)
LC	0.8	0.5	0.7	0.1
~LC	0.2	0.5	0.3	0.9

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:

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

Training Bayesian Networks

- Several scenarios:
  - Given both the network structure and all variables observable: *learn only the CPTs*
  - Network structure known, some hidden variables: *gradient descent* (greedy hill-climbing) method, analogous to neural network learning
  - Network structure unknown, all variables observable: search through the model space to *reconstruct network topology*
  - Unknown structure, all hidden variables: No good algorithms known for this purpose

- Ref. D. Heckerman: Bayesian networks for data mining