Bayesian Learning

Kaiqi Zhao
The University of Auckland

Slides are partially based on the materials from Mitchel's book and Stanford's NLP lectures

Bayes Optimal Classifier

Bayes Optimal Classifier

- Question: What is the most probable classification of the new instance given the training data?
- Simply applying h_{MAP} is not the best solution (as one could wrongly think of)
- Example
  - $H = \{h1, h2, h3\}$, where $P(h_1|D) = 0.4, \quad P(h_2|D) = P(h_3|D) = 0.3$
  - $h_{MAP} = h_1$
  - Consider a new instance x encountered, which is classified positive by h_1 and negative by h_2 , h_3
  - Taking all hypotheses into account
    - The probability that x is positive is 0.4 (the posterior probability of h_1)
    - The probability that x is negative is 0.6

⇒ most probable classification is not the classification result generated by h_{MAP}

Bayes Optimal Classifier

- The most probable classification is obtained by combining the predictions of all hypotheses, weighted by their posterior probabilities

$$P(v_j|D) = \sum_{h_i \in H} P(v_j|h_i)P(h_i|D)$$

where $P(v_j|D)$ is the probability that the correct classification is v_j

- Bayes optimal classifier

$$\text{argmax}_{v_j \in V} P(v_j|D) = \text{argmax}_{v_j \in V} \sum_{h_i \in H} P(v_j|h_i)P(h_i|D)$$

Example

$V = \{\oplus, \ominus\}$

$$P(h_1|D) = 0.4, P(\ominus|h_1) = 0, P(\oplus|h_1) = 1$$

$$P(h_2|D) = 0.3, P(\ominus|h_2) = 1, P(\oplus|h_2) = 0$$

$$P(h_3|D) = 0.3, P(\ominus|h_3) = 1, P(\oplus|h_3) = 0$$

therefore

$$P(\oplus|D) = \sum_{h_i \in H} P(\oplus|h_i)P(h_i|D) = 0.4$$

$$P(\ominus|D) = \sum_{h_i \in H} P(\ominus|h_i)P(h_i|D) = 0.6$$

and

$$\text{argmax}_{v_j \in \{\oplus, \ominus\}} \sum_{h_i \in H} P(v_j|h_i)P(h_i|D) = \ominus$$

Naive Bayes Classifier

Naive Bayes Classifier

- Applies to learning tasks where each instance x is described by a conjunction of attribute values $< a_1, a_2, \ldots, a_n >$ and where the target function $f(x)$ can take any value from some finite set V
- Maximum A Posteriori

$$v_{MAP} = \text{argmax}_{v_j \in V} P(v_j | a_1, a_2, \ldots, a_n)$$

$$= \text{argmax}_{v_j \in V} \frac{P(a_1, a_2, \ldots, a_n | v_j) P(v_j)}{P(a_1, a_2, \ldots, a_n)}$$

$$= \text{argmax}_{v_j \in V} P(a_1, a_2, \ldots, a_n | v_j) P(v_j)$$

Naive Bayes Classifier

- $P(v_j)$ can be estimated by counting the frequency of v_j in D
- $P(a_1, a_2, \ldots, a_n | v_j)$ cannot be estimated in this fashion
- Number of these terms is |all possible instances| $\times$ $|V|$
- Naive Bayes classifier
  - **Assumption: attribute values are conditionally independent given the target value**

$$P(a_1, a_2, \ldots, a_n | v_j) = \prod_i P(a_i | v_j)$$

  - Hence, number of terms is |distinct attribute values| $\times$ $|V| + |V|$
  - No explicit search through H, just counting frequencies

Maximum A Posteriori of the prediction target ⇒ Naive Bayes Classifier

$$v_{NB} = \text{argmax}_{v_j \in V} P(v_j) \prod_i P(a_i | v_j)$$

Naive Bayes for Document Classification

- For a document d and a class c

$$P(c|d) = \frac{P(d|c)P(c)}{P(d)}$$

- Using MAP:

$$C_{MAP} = \text{argmax}_{c \in C} P(c|d)$$

$$= \text{argmax}_{c \in C} \frac{P(d|c)P(c)}{P(d)}$$

$$= \text{argmax}_{c \in C} P(d|c)P(c)$$

$$= \text{argmax}_{c \in C} P(x_1, x_2, \ldots, x_n|c)P(c)$$

Naive Bayes for Document Classification

$$C_{MAP} = \text{argmax}_{c \in C} P(x_1, x_2, \ldots, x_n | c) P(c)$$

- Each document is presented by features $x_1, x_2, \ldots, x_n$
- Let X be the set of unique feature values (e.g. unique words)
- Assuming the features are independent

$$C_{MAP} = \text{argmax}_{c \in C} P(x_1 | c) \cdot P(x_2 | c) \cdot \cdots \cdot P(x_n | c) P(c)$$

$$= \text{argmax}_{c \in C} P(c) \prod_i P(x_i | c)$$

Naive Bayes for Document Classification

$$C_{MAP} = \text{argmax}_{c \in C} P(c) \prod_i P(x_i|c)$$

- $P(c)$ is easy to estimate – you can count the frequency of the class in your dataset
- How about $P(x_i|c)$?
- Based on the co-occurrence of word x_i and class c

$$P(x_i|c) = \frac{count(x_i, c)}{\sum_{x_j \in X} count(x_j, c)}$$

Laplace smoothing

$$P(x_i|c) = \frac{count(x_i, c)}{\sum_{x_j \in X} count(x_j, c)}$$

- What if $\sum_{x_j \in X} count(x_j, c) = 0$?
- Solution: Simply add a constant

$$P(x_i|c) = \frac{count(x_i, c) + 1}{\sum_{x_j \in X} (count(x_j, c) + 1)}$$

A very simple Naive Bayes Example

	Doc_ID	Words	class
Training	d1	Kiwi, Sheep, Kiwi	NZ
	d2	Kiwi,Kiwi,Bird	NZ
	d3	Kiwi,Auckland	NZ
	d4	Munich,Oktoberfest,Kiwi	DE
Test	d5	Kiwi,Kiwi,Kiwi,Munich,Oktoberfest	NZ?

$$P(c) = \frac{N_c}{N}$$

$$P(x|c) = \frac{count(x,c) + 1}{\sum_{x_j \in X} count(x_j, c) + |X|}$$

- Priors

$$P(NZ) = 3/4$$
$$P(DE) = 1/4$$

- Choosing a class for $d5$

$$P(NZ|d5) \propto 3/4 * (3/7)^3 * 1/14 * 1/14 \approx 0.0003$$

$$P(DE|d5) \propto 1/4 * (2/9)^3 * 2/9 * 2/9 \approx 0.0001$$

- Conditional Probabilities

$$P(Kiwi|NZ) = (5+1)/(8+6) = 3/7$$
$$P(Munich|NZ) = (0+1)/(8+6) = 1/14$$
$$P(Oktoberfest|NZ) = (0+1)/(8+6) = 1/14$$
$$P(Kiwi|DE) = (1+1)/(3+6) = 2/9$$
$$P(Munich|DE) = (1+1)/(3+6) = 2/9$$
$$P(Oktoberfest|DE) = (1+1)/(3+6) = 2/9$$

Bayesian Networks

Bayesian Networks

- Naive Bayes assumption of conditional independence too restrictive
- But it's intractable without such assumptions
- Bayesian Belief networks describe conditional independence among subsets of variables
- Allows combining prior knowledge about (in)dependencies among variables with observed training data

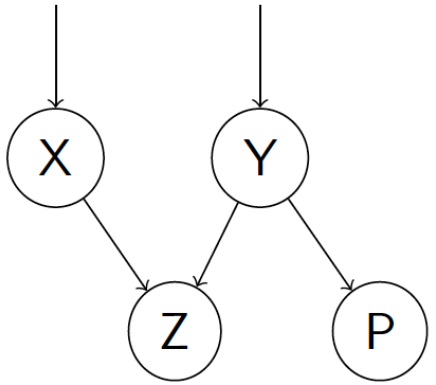
Bayesian Networks

- A simple, graphical notation for conditional independence assertions and hence for compact specification of full joint distributions
- Syntax:
  - A set of nodes, one per variable
  - A directed, acyclic graph (link ≈ "directly influences")
  - A conditional distribution for each node given its parents: $P(X_i|Parents(X_i))$
- Conditional distribution represented as a conditional probability table (CPT) giving the distribution of X_i for each combination of parent values

Bayesian Networks – A Graphical Model

- A graphical model of relationships
  - Represents dependencies among the variables
    - **Can** be causal dependencies
  - Gives a specification of joint probability distribution:

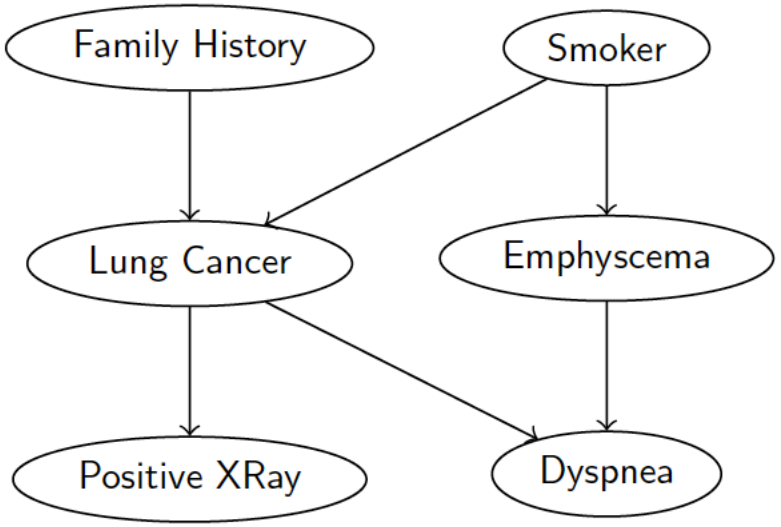
$$P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P(X_i | Parents(X_i))$$



- Nodes: random variables
- Links: dependencies
- X, Y are parents of Z, and Y is the parent of P
- Has no loops or cycles

$$P(X, Y, Z, P) = P(Z|X, Y)P(P|Y)P(X)P(Y)$$

Bayesian Belief Network – Example

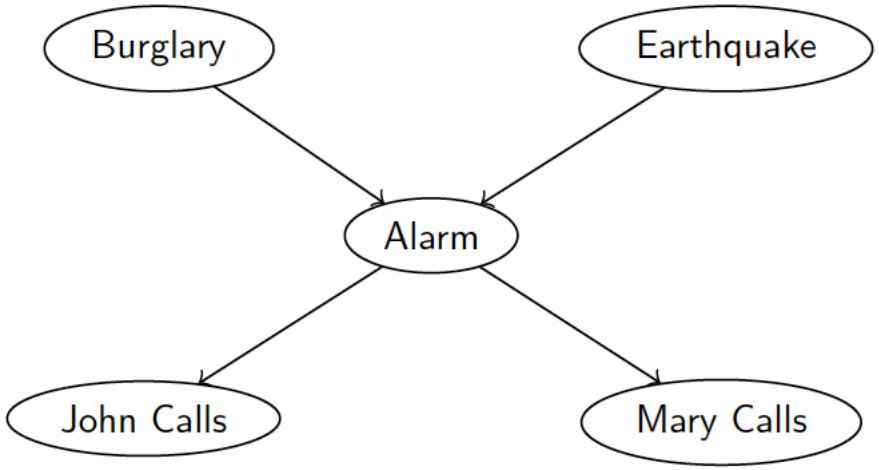


	$FH = T$ $S = T$	$FH = T$ $S = F$	$FH = F$ $S = T$	$FH = F$ $S = F$
$LC = T$	0.8	0.5	0.7	0.1
$LC = F$	0.2	0.5	0.3	0.9

- The conditional probability table for the variable Lung Cancer $P(LC|FH, S)$
- Shows the conditional probability for each possible combination of its parents

$$P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P(X_i|Parents(X_i))$$

Bayesian Belief Network – Another Example



P(B)	P(¬B)	P(E)	P(¬E)
0.001	0.999	0.002	0.998

B	E	$P(A)$	$P(\neg A)$
T	T	0.95	0.05
T	F	0.94	0.06
F	T	0.29	0.71
F	F	0.001	0.999

A	P(J)	P(¬J)
T	0.9	0.1
F	0.05	0.95

A	P(M)	P(¬M)
T	0.7	0.3
F	0.01	0.99

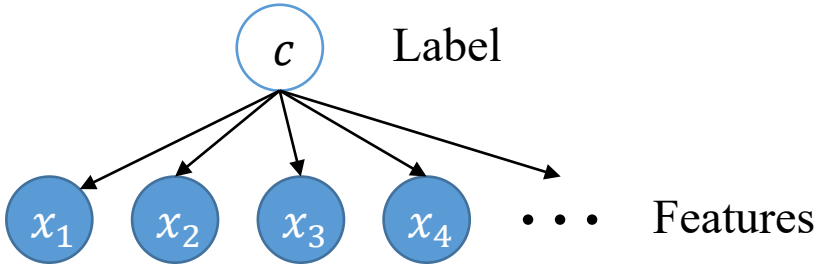
- What is the probability that an alarm has sounded, both Mary and John call, and there was no earthquake and no burglary?

$$P(A, J, M, \neg B, \neg E) = P(J|A) \cdot P(M|A) \cdot P(A|\neg B, \neg E) \cdot P(\neg B) \cdot P(\neg E)$$

$$= 0.9 \cdot 0.7 \cdot 0.001 \cdot 0.999 \cdot 0.998 = 0.00062$$

Application – Naïve Bayes Classifier

- Naïve Bayes classifier is actually a special case of Bayesian network
  - Variables → features x_i, class c
  - Graph structure:



$$P(x_1, .., x_n, c) = P(c) \prod_i P(x_i|c)$$

Application – Gaussian Mixture Model

- Consider a model that clusters data points $X \in \mathbb{R}^{N \times d}$ in a continuous space into K groups
  - Variables → data point x, group g
  - Graph structure:

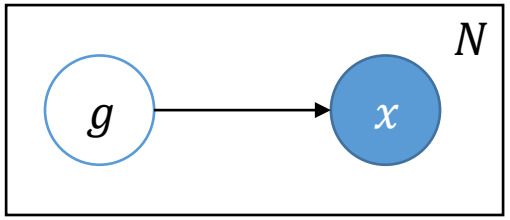
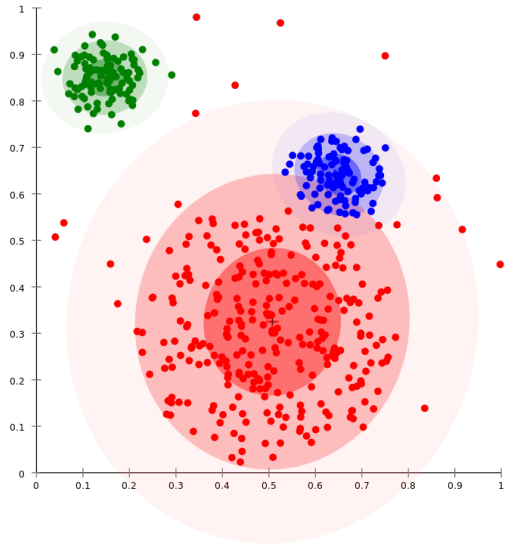


Plate notation

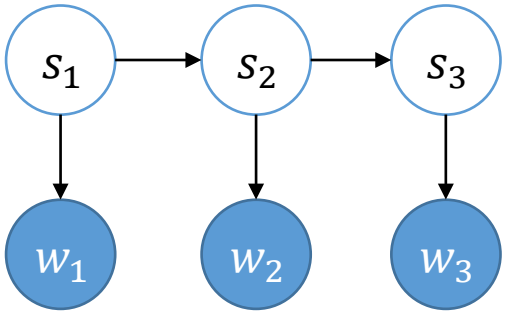


  - Probability distribution:
    - $p(g) \sim Categorical(K)$
    - $p(x|g) \sim Normal(\mu_g, \Sigma_g)$

$$p\,(X, G) = \prod_{g} p(g) \prod_{i=1}^{N} p(x_i|g_i)$$

Application – Hidden Markov Model

- Consider a model that predicts tomorrow's weather given the weather condition so far.
  - Variables → state s (hidden); weather w (observed)
  - Graph structure:



  - Discrete Distribution:
    - $p(s)$
    - $p(s_i|s_{i-1})$
    - $p(w|s)$

$$
\begin{aligned}
& p(W,S) = p(w_1,\dots,w_T,s_1,\dots,s_T) \\
= \; & p(s_1)\,p(w_1|s_1)\,p(s_2|s_1)\,p(w_2|s_2) \;\dots\; p(s_T|s_{T-1})\,p(w_T|s_T) \\
= \; & p(s_1)\,p(s_2|s_1) \;\dots\; p(s_T|s_{T-1}) \\
\times \; & p(w_1|s_1)\,p(w_2|s_2) \;\dots\; p(w_T|s_T)
\end{aligned}
$$

Inference in Bayesian Networks

- We are often interested in the unobservable (hidden) variables, e.g., the hidden states in HMM, the cluster label in GMM.
- If only one variable with unknown value (e.g. target in classification), easy to infer it, e.g., MAP or ML
- In general case, the exact inference is NP-hard, approximations have been introduced (e.g. Monte Carlo-based methods and variational inference)
- In some cases, the structure is unknown, we may need to learn the structure as well!
- Refer to More on Bayesian Networks in Bishop's *Pattern Recognition And Machine Learning*

Summary

- Bayesian learning relies on Bayes' Theorem
- Bayesian methods can be used to select the most likely hypothesis (MAP/ML) given the data
- Bayesian Learning has multiple roles
  - Provide practical and effective learning algorithms like Naive Bayes
  - Provide a framework
    - For evaluating other learners
    - For analyzing learning
- Bayes optimal classifier combines the predictions of all alternative hypothesis weighted by their posterior probabilities
- Bayesian networks provide a natural representation for conditional independence

Literature

- Chapter 8 of Bishop's Pattern *Recognition and Machine Learning*
- Chapter 6 of Mitchell's *Machine Learning*