CS685: Data Mining
Bayesian Classifiers

Arnab Bhattacharya
arnabb@cse.iitk.ac.in

Computer Science and Engineering,
Indian Institute of Technology, Kanpur
http://web.cse.iitk.ac.in/~cs685/

1st semester, 2018-19

Mon, Thu 1030-1145 at RM101

Bayes' Theorem

$$P(C|O) = \frac{P(O|C)P(C)}{P(O)}$$

- $P(C|O)$ is the probability of class C given object O – posterior probability
- $P(O|C)$ is the probability that O is from class C – likelihood probability
- $P(C)$ is the probability of class C – prior probability
- $P(O)$ is the probability of object O – evidence probability

$$posterior = \frac{likelihood \times prior}{evidence}$$

Naïve Bayes Classifier

- Naïve Bayes classifier or Simple Bayes classifier
- To classify a new object O_q, compute posterior probabilities $P(C_i|O_q)$ for all classes $C_i, i = 1, \ldots, k$

$$P(C_i|O_q) = \frac{P(O_q|C_i)P(C_i)}{P(O_q)}$$

- Bayes decision rule: The class with the *highest* posterior probability is chosen

Naïve Bayes Classifier

- Naïve Bayes classifier or Simple Bayes classifier
- To classify a new object O_q, compute posterior probabilities $P(C_i|O_q)$ for all classes $C_i, i = 1, \dots, k$

$$P(C_i|O_q) = \frac{P(O_q|C_i)P(C_i)}{P(O_q)}$$

- Bayes decision rule: The class with the *highest* posterior probability is chosen
- $P(O_q)$ is constant for all classes and, therefore, can be removed

Naïve Bayes Classifier

- Naïve Bayes classifier or Simple Bayes classifier
- To classify a new object O_q, compute posterior probabilities $P(C_i|O_q)$ for all classes $C_i, i = 1, \dots, k$

$$P(C_i|O_q) = \frac{P(O_q|C_i)P(C_i)}{P(O_q)}$$

- Bayes decision rule: The class with the *highest* posterior probability is chosen
- $P(O_q)$ is constant for all classes and, therefore, can be removed
- Since it maximizes posterior probability, it is called maximum a posteriori (MAP) method

Naïve Bayes Classifier

- Naïve Bayes classifier or Simple Bayes classifier
- To classify a new object O_q, compute posterior probabilities $P(C_i|O_q)$ for all classes $C_i, i = 1, \dots, k$

$$P(C_i|O_q) = \frac{P(O_q|C_i)P(C_i)}{P(O_q)}$$

- Bayes decision rule: The class with the *highest* posterior probability is chosen
- $P(O_q)$ is constant for all classes and, therefore, can be removed
- Since it maximizes posterior probability, it is called maximum a posteriori (MAP) method
- If priors are unknown or same, this essentially maximizes the likelihood $P(O_q|C_i)$
- This is called maximum likelihood (ML) method

Computing Likelihood

- In general, O_q has m features $O_q = \langle O_{q_1}, \dots, O_{q_m} \rangle$

$$
\begin{aligned}
P(O_q|C_i) &= P(O_{q_1}, O_{q_2}, \dots, O_{q_m}|C_i) \\
&= P(O_{q_1}|C_i) \times P(O_{q_2}, \dots, O_{q_m}|O_{q_1}, C_i) \\
&= P(O_{q_1}|C_i) \times P(O_{q_2}|O_{q_1}, C_i) \times P(O_{q_3}, \dots, O_{q_m}|O_{q_1}, O_{q_2}, C_i)
\end{aligned}
$$

Computing Likelihood

- In general, O_q has m features $O_q = \langle O_{q_1}, \ldots, O_{q_m} \rangle$

$$
\begin{aligned}
P(O_q|C_i) &= P(O_{q_1}, O_{q_2}, \ldots, O_{q_m}|C_i) \\
&= P(O_{q_1}|C_i) \times P(O_{q_2}, \ldots, O_{q_m}|O_{q_1}, C_i) \\
&= P(O_{q_1}|C_i) \times P(O_{q_2}|O_{q_1}, C_i) \times P(O_{q_3}, \ldots, O_{q_m}|O_{q_1}, O_{q_2}, C_i)
\end{aligned}
$$

- *Simple* or *naïve* assumption is now applied: All class conditional probabilities are independent

Computing Likelihood

- In general, O_q has m features $O_q = \langle O_{q_1}, \dots, O_{q_m} \rangle$

$$\begin{aligned} P(O_q|C_i) &= P(O_{q_1}, O_{q_2}, \dots, O_{q_m}|C_i) \\ &= P(O_{q_1}|C_i) \times P(O_{q_2}, \dots, O_{q_m}|O_{q_1}, C_i) \\ &= P(O_{q_1}|C_i) \times P(O_{q_2}|O_{q_1}, C_i) \times P(O_{q_3}, \dots, O_{q_m}|O_{q_1}, O_{q_2}, C_i) \end{aligned}$$

- *Simple* or *naïve* assumption is now applied: All class conditional probabilities are independent

$$\begin{aligned} P(O_{q_j}, O_{q_k}|C_i) &= P(O_{q_j}|C_i) \times P(O_{q_k}|O_{q_j}, C_i) \\ &= P(O_{q_j}|C_i) \times P(O_{q_k}|C_i) \quad [\because O_{q_j}, O_{q_k} \text{ are independent}] \end{aligned}$$

Computing Likelihood

- In general, O_q has m features $O_q = \langle O_{q_1}, \dots, O_{q_m} \rangle$

$$\begin{aligned} P(O_q|C_i) &= P(O_{q_1}, O_{q_2}, \dots, O_{q_m}|C_i) \\ &= P(O_{q_1}|C_i) \times P(O_{q_2}, \dots, O_{q_m}|O_{q_1}, C_i) \\ &= P(O_{q_1}|C_i) \times P(O_{q_2}|O_{q_1}, C_i) \times P(O_{q_3}, \dots, O_{q_m}|O_{q_1}, O_{q_2}, C_i) \end{aligned}$$

- *Simple* or *naïve* assumption is now applied: All class conditional probabilities are independent

$$\begin{aligned} P(O_{q_j}, O_{q_k}|C_i) &= P(O_{q_j}|C_i) \times P(O_{q_k}|O_{q_j}, C_i) \\ &= P(O_{q_j}|C_i) \times P(O_{q_k}|C_i) \quad [\because O_{q_j}, O_{q_k} \text{ are independent}] \end{aligned}$$

$$\therefore P(O_q|C_i) = P(O_{q_1}|C_i) \times P(O_{q_2}|C_i) \times P(O_{q_3}, \dots, O_{q_m}|O_{q_1}, O_{q_2}, C_i)$$

$$\text{or, } P(O_q|C_i) = P(O_{q_1}, O_{q_2}, \dots, O_{q_m}|C_i) = \prod_{j=1}^{m} P(O_{q_j}|C_i)$$

Training

- How to estimate $P(O_{q_j}|C_i)$

Training

- How to estimate $P(O_{q_j}|C_i)$
- Examine all training objects pertaining to class C_i

Training

- How to estimate $P(O_{q_j}|C_i)$
- Examine all training objects pertaining to class C_i
- If O_{q_j} is categorical, then relative empirical frequencies are estimates

$$P(O_{q_j} = v|C_i) = \frac{|\{O_k \in C_i : O_{k_j} = v\}|}{|\{O_k \in C_i\}|}$$

- A particular discrete distribution can also be assumed

Training

- How to estimate $P(O_{q_j}|C_i)$
- Examine all training objects pertaining to class C_i
- If O_{q_j} is categorical, then relative empirical frequencies are estimates

$$P(O_{q_j} = v|C_i) = \frac{|\{O_k \in C_i : O_{k_j} = v\}|}{|\{O_k \in C_i\}|}$$

- A particular discrete distribution can also be assumed
- If O_{q_j} is numerical, then a certain continuous distribution is assumed
- Generally, Gaussian or normal distribution $N(\mu, \sigma)$
- μ and σ are estimated from training objects in C_i

$$P(O_{q_j} = v|C_i) = N(v; \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(v-\mu)^2}{2\sigma^2}}$$

Training

- How to estimate $P(O_{q_j}|C_i)$
- Examine all training objects pertaining to class C_i
- If O_{q_j} is categorical, then relative empirical frequencies are estimates

$$P(O_{q_j} = v|C_i) = \frac{|\{O_k \in C_i : O_{k_j} = v\}|}{|\{O_k \in C_i\}|}$$

- A particular discrete distribution can also be assumed
- If O_{q_j} is numerical, then a certain continuous distribution is assumed
- Generally, Gaussian or normal distribution $N(\mu, \sigma)$
- μ and σ are estimated from training objects in C_i

$$P(O_{q_j} = v|C_i) = N(v; \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(v-\mu)^2}{2\sigma^2}}$$

- $P(C_i)$ is just the empirical estimate $|C_i|/|D|$

Example: training

Class	Rank	Motivated	Exam marks
Successful (S)	2	Y	78.3
	99	Y	70.3
	5	N	88.5
	87	Y	75.1
Unsuccessful (U)	1	N	76.3
	90	N	66.2
	9	Y	68.1
	62	N	75.4

Example: training

Class	Rank	Motivated	Exam marks
Successful (S)	2	Y	78.3
	99	Y	70.3
	5	N	88.5
	87	Y	75.1
Unsuccessful (U)	1	N	76.3
	90	N	66.2
	9	Y	68.1
	62	N	75.4

Likelihoods

Class	Rank	Motivated	Exam marks
S	$\mu = 48.25$ $\sigma = 51.92$	$P(Y) = 0.75$ $P(N) = 0.25$	$\mu = 78.05$ $\sigma = 7.70$
U	$\mu = 40.50$ $\sigma = 42.68$	$P(Y) = 0.25$ $P(N) = 0.75$	$\mu = 71.50$ $\sigma = 5.10$

Example: testing

- $O_q = (70, Y, 67.3)$

Example: testing

- $O_q = (70, Y, 67.3)$

$$\begin{aligned}
P(O_q|S) &= P(70|S) \times P(Y|S) \times P(67.3|S) \times P(S) \\
&= N(70; 48.25, 51.92) \times 0.75 \times N(67.3; 78.05, 7.70) \times 0.5 \\
&= 0.00704 \times 0.75 \times 0.0195 \times 0.5 \\
&= 5.16 \times 10^{-5} \\
P(O_q|U) &= P(70|U) \times P(Y|U) \times P(67.3|U) \times P(U) \\
&= N(70; 40.50, 42.68) \times 0.25 \times N(67.3; 71.50, 5.10) \times 0.5 \\
&= 0.00736 \times 0.25 \times 0.0597 \times 0.5 \\
&= 5.49 \times 10^{-5}
\end{aligned}$$

- Therefore, O_q is from class U

Discussion

- If an estimated probability $P(O_{q_j}|C_i)$ becomes zero, the whole likelihood becomes zero
- Laplacian correction or Laplacian estimation: Add a small ϵ

Discussion

- If an estimated probability $P(O_{q_j}|C_i)$ becomes zero, the whole likelihood becomes zero
- Laplacian correction or Laplacian estimation: Add a small ϵ
- Advantages

Discussion

- If an estimated probability $P(O_{q_j}|C_i)$ becomes zero, the whole likelihood becomes zero
- Laplacian correction or Laplacian estimation: Add a small ϵ
- Advantages
  - Incremental

Discussion

- If an estimated probability $P(O_{q_j}|C_i)$ becomes zero, the whole likelihood becomes zero
- Laplacian correction or Laplacian estimation: Add a small ϵ
- Advantages
  - Incremental
  - Robust to noise

Discussion

- If an estimated probability $P(O_{q_j}|C_i)$ becomes zero, the whole likelihood becomes zero
- Laplacian correction or Laplacian estimation: Add a small ϵ
- Advantages
  - Incremental
  - Robust to noise as probability of noise is low

Discussion

- If an estimated probability $P(O_{q_j}|C_i)$ becomes zero, the whole likelihood becomes zero
- Laplacian correction or Laplacian estimation: Add a small ϵ
- Advantages
  - Incremental
  - Robust to noise as probability of noise is low
  - Robust to irrelevant attributes

Discussion

- If an estimated probability $P(O_{q_j}|C_i)$ becomes zero, the whole likelihood becomes zero
- Laplacian correction or Laplacian estimation: Add a small ϵ
- Advantages
  - Incremental
  - Robust to noise as probability of noise is low
  - Robust to irrelevant attributes as their probability tends to be uniform across classes
- Disadvantages

Discussion

- If an estimated probability $P(O_{q_j}|C_i)$ becomes zero, the whole likelihood becomes zero
- Laplacian correction or Laplacian estimation: Add a small ϵ
- Advantages
  - Incremental
  - Robust to noise as probability of noise is low
  - Robust to irrelevant attributes as their probability tends to be uniform across classes
- Disadvantages
  - Treats attributes as independent and ignores any correlation information
  - Two redundant attributes contribute twice the weight

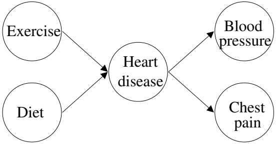
Bayesian Networks

- Bayesian networks or Bayesian belief networks or Bayes nets or belief nets
- Takes into account the correlations of attributes by modeling them as conditional probabilities
- Forms a directed acyclic graph (DAG)
- Edges model the *dependencies*
- Parent is the *cause* and children are the *effects*

Bayesian Networks

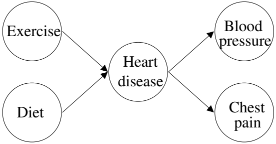
- Bayesian networks or Bayesian belief networks or Bayes nets or belief nets
- Takes into account the correlations of attributes by modeling them as conditional probabilities
- Forms a directed acyclic graph (DAG)
- Edges model the *dependencies*
- Parent is the *cause* and children are the *effects*
- A node is conditionally independent of all its non-descendants given its parents
- For every node, there is a conditional probability table (CPT) that describes its values given its parents' values
- CPT for node X is of the form $P(X|parents(X))$

Example



- CPTs: rows are values; columns are parents (i.e., conditionals)
- Last rows can be inferred, and therefore, omitted

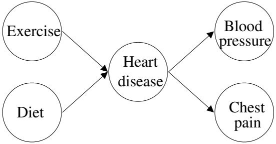
Example



- CPTs: rows are values; columns are parents (i.e., conditionals)
- Last rows can be inferred, and therefore, omitted

Exercise (E)	Φ
regular (r)	0.70
irregular (i)	0.30

Example

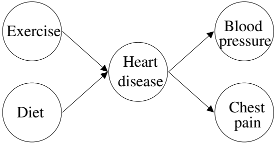


- CPTs: rows are values; columns are parents (i.e., conditionals)
- Last rows can be inferred, and therefore, omitted

Exercise (E)	Φ
regular (r)	0.70
irregular (i)	0.30

Diet (D)	Φ
healthy (h)	0.25
unhealthy (u)	0.75

Example



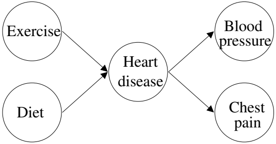
- CPTs: rows are values; columns are parents (i.e., conditionals)
- Last rows can be inferred, and therefore, omitted

Exercise (E)	Φ
regular (r)	0.70
irregular (i)	0.30

Diet (D)	Φ
healthy (h)	0.25
unhealthy (u)	0.75

Heart disease (H)	E=r, D=h	E=r, D=u	E=i, D=h	E=i, D=u
yes (y)	0.25	0.40	0.55	0.80
no (n)	0.75	0.60	0.45	0.20

Example



- CPTs: rows are values; columns are parents (i.e., conditionals)
- Last rows can be inferred, and therefore, omitted

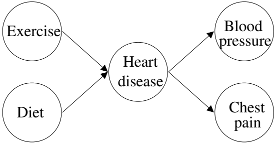
Exercise (E)	Φ
regular (r)	0.70
irregular (i)	0.30

Diet (D)	Φ
healthy (h)	0.25
unhealthy (u)	0.75

Heart disease (H)	E=r, D=h	E=r, D=u	E=i, D=h	E=i, D=u
yes (y)	0.25	0.40	0.55	0.80
no (n)	0.75	0.60	0.45	0.20

Blood pressure (B)	H=y	H=n
normal (l)	0.15	0.80
high (g)	0.85	0.20

Example



- CPTs: rows are values; columns are parents (i.e., conditionals)
- Last rows can be inferred, and therefore, omitted

Exercise (E)	Φ
regular (r)	0.70
irregular (i)	0.30

Diet (D)	Φ
healthy (h)	0.25
unhealthy (u)	0.75

Heart disease (H)	E=r, D=h	E=r, D=u	E=i, D=h	E=i, D=u
yes (y)	0.25	0.40	0.55	0.80
no (n)	0.75	0.60	0.45	0.20

Blood pressure (B)	H=y	H=n
normal (l)	0.15	0.80
high (g)	0.85	0.20

Chest pain (C)	H=y	H=n
normal (m)	0.70	0.45
pain (p)	0.30	0.55

Classification using Bayesian Networks

- Given no prior information, is a person suffering from heart disease?
- Essentially, a yes/no classification problem with some information
- Note that no other information (e.g., chest pain, etc.) are known
- Compute $P(H = y)$; if it is greater than $P(H = n)$, then predict "heart disease"

Classification using Bayesian Networks

- Given no prior information, is a person suffering from heart disease?
- Essentially, a yes/no classification problem with some information
- Note that no other information (e.g., chest pain, etc.) are known
- Compute $P(H = y)$; if it is greater than $P(H = n)$, then predict "heart disease"

$$
\begin{aligned}
P(H = y) &= \sum_{\alpha,\beta} [P(H = y|E = \alpha, D = \beta).P(E = \alpha, D = \beta)] \\
&= \sum_{\alpha,\beta} [P(H = y|E = \alpha, D = \beta).P(E = \alpha).P(D = \beta)] \\
&= 0.25 \times 0.70 \times 0.25 + 0.40 \times 0.70 \times 0.75 \\
&\quad\quad + 0.55 \times 0.30 \times 0.25 + 0.80 \times 0.30 \times 0.75 \\
&= 0.475
\end{aligned}
$$

Classification using Bayesian Networks (contd.)

- Given a person has high blood pressure, is she suffering from heart disease?
- Essentially, a yes/no classification problem with some information
- Note that not all information (e.g., chest pain, etc.) are known
- Compute $P(H = y|B = g)$; if it is greater than $P(H = n|B = g)$, then predict "heart disease"

Classification using Bayesian Networks (contd.)

- Given a person has high blood pressure, is she suffering from heart disease?
- Essentially, a yes/no classification problem with some information
- Note that not all information (e.g., chest pain, etc.) are known
- Compute $P(H = y|B = g)$; if it is greater than $P(H = n|B = g)$, then predict "heart disease"

$$
\begin{aligned}
P(H = y|B = g) &= \frac{P(B = g|H = y).P(H = y)}{P(B = g)} \\
&= \frac{P(B = g|H = y).P(H = y)}{\sum_\alpha [P(B = g|H = \alpha).P(H = \alpha)]} \\
&= \frac{0.85 \times 0.475}{0.85 \times 0.475 + 0.20 \times 0.525} \\
&= 0.794
\end{aligned}
$$

Classification using Bayesian Networks (contd.)

- Given a person has high blood pressure, unhealthy diet and irregular exercise, is she suffering from heart disease?
- Essentially, a yes/no classification problem with some information
- Note that not all information (e.g., chest pain, etc.) are known
- Compute $P(H = y | B = g, D = u, E = i)$; if it is greater than $P(H = n | B = g, D = u, E = i)$, then predict "heart disease"

Classification using Bayesian Networks (contd.)

- Given a person has high blood pressure, unhealthy diet and irregular exercise, is she suffering from heart disease?
- Essentially, a yes/no classification problem with some information
- Note that not all information (e.g., chest pain, etc.) are known
- Compute $P(H = y|B = g, D = u, E = i)$; if it is greater than $P(H = n|B = g, D = u, E = i)$, then predict "heart disease"

$$
\begin{aligned}
P(H = y|B = g, D = u, E = i) \\
&= \frac{P(B = g|H = y, D = u, E = i).P(H = y|D = u, E = i)}{P(B = g|D = u, E = i)} \\
&= \frac{P(B = g|H = y).P(H = y|D = u, E = i)}{\sum_{\alpha} [P(B = g|H = \alpha).P(H = \alpha|D = u, E = i)]} \\
&= \frac{0.85 \times 0.80}{0.85 \times 0.80 + 0.20 \times 0.20} \\
&= 0.944
\end{aligned}
$$

Training

- Two important steps

Training

- Two important steps
- Learning the network topology
  - Which edges are present?

Training

- Two important steps
- Learning the network topology
  - Which edges are present?
  - Domain knowledge from human experts

Training

- Two important steps
- Learning the network topology
  - Which edges are present?
  - Domain knowledge from human experts
- Learning the CPTs

Training

- Two important steps
- Learning the network topology
  - Which edges are present?
  - Domain knowledge from human experts
- Learning the CPTs
  - Same method as naïve Bayes
  - Empirical probabilities
  - If not categorical, use Gaussian

Discussion

- Models reality better

Discussion

- Models reality better
- Dependence or correlation does not indicate which is cause and which is effect
  - Class is boring and not paying attention

Discussion

- Models reality better
- Dependence or correlation does not indicate which is cause and which is effect
    - Class is boring and not paying attention
- Topology of network is very important

Discussion

- Models reality better
- Dependence or correlation does not indicate which is cause and which is effect
  - Class is boring and not paying attention
- Topology of network is very important
- For large CPTs, require lots of training data

Discussion

- Models reality better
- Dependence or correlation does not indicate which is cause and which is effect
    - Class is boring and not paying attention
- Topology of network is very important
- For large CPTs, require lots of training data
- Naïve Bayes is a special case

Discussion

- Models reality better
- Dependence or correlation does not indicate which is cause and which is effect
  - Class is boring and not paying attention
- Topology of network is very important
- For large CPTs, require lots of training data
- Naïve Bayes is a special case
  - Class is parent and attributes are children