Discrete Bayesian classifiers

Lecture 5

Outline

- Bayes theorem
- Maximum likelihood classification
- “Brute force” Bayesian learning
- Naïve Bayes
- Bayesian Belief Networks

Bayes theorem

$$P(c \mid x) = \frac{P(x \mid c)P(c)}{P(x)}$$

- P(c) – prior probability of class c
  - Expected proportion of data from class c on test
- P(x) – prior probability of instance x
  - Probability of instance with attribute vector x to occur
- P(c|x) – probability of an instance being of class c given it is described by vector of attributes x
- P(x|c) – probability of an instance of having attributes described by x given it comes from class c

Maximum likelihood

- From training data estimate for all x and c_i
  - $P(c_i)$
  - $P(x|c_i)$
- During classification:
  - Choose class c_i with maximal $P(c_i|x)$
- $P(c_i|x)$ is also called likelihood

‘Brute force’ Bayesian learning

- Instance x described by attributes <$a_1,\dots,a_n$>
- Most probable class:

$$c(x) = \arg\max_{c_i} P(c_i \mid a_1,\dots,a_n) =$$

$$= \arg\max_{c_i} \frac{P(a_1,\dots,a_n \mid c_i)P(c_i)}{P(a_1,\dots,a_n)} =$$

$$= \arg\max_{c_i} P(a_1,\dots,a_n \mid c_i)P(c_i)$$

Example

- Consider Data

Wind	Rain	Balloon
Strong	Shower	No
Strong	Shower	No
Strong	None	No
Weak	None	Yes
Weak	Shower	No
Weak	None	Yes
Weak	None	No
Weak	None	Yes

- We estimate:
  - P(No) = 62%
  - P(Yes) = 38%
  - P(<Weak,None>|No) = 20%
  - P(<Weak,None>|Yes)=100%
  - P(<Weak,Show.>|No) = 20%
  - …
- Classification of <Weak,None>
  - L(No) ~ 0.62 x 0.2= 0.12
  - L(Yes) ~ 0.38 x 1= 0.38
  - Answer: Yes

Problem with ‘Brute force’

- It cannot generalize to unseen examples x^{new}, because it does not have estimates $P(c_i|x^{new})$
- It is useless
- Brute force does not have any bias
- So in order to make learning possible we have to introduce a bias

Naïve Bayes

- Brute force: $c(x) = \arg\max_{c_i} P(a_1,\dots,a_n \mid c_i)P(c_i)$
- Naïve Bayes assumes that attributes are independent for instances from a given class:

$$P(a_1,\dots,a_n \mid c_i) = \prod_j P(a_j \mid c_i)$$

- Which gives: $\boxed{c(x) = \arg\max_{c_i} P(c_i)\prod_j P(a_j \mid c_i)}$
  - Assumption of independence is often violated by Naïve Bayes works surprisingly well anyway

Example

- Recall ‘advanced ballooning’ set:

Sky	Temper.	Rain	Wind	Fly Balloon
Sunny	Cold	None	Strong	Yes
Cloudy	Cold	Shower	Weak	Yes
Cloudy	Cold	Shower	Strong	No
Sunny	Hot	Shower	Strong	No

- Classify: x=<Cloudy, Hot, Shower, Strong>
  - P(Y|x) ~ P(Y) P(Cl|Y) P(H|Y) P(Sh|Y) P (St|Y)
    = 0.5 x 0.5 x 0 x 0.5 x 0.5 = 0
  - P(N|x) ~ 0.5 x 0.5 x 0.5 x 1 x 1 = 0.125

Missing estimates

- What if none of training instances of class c_i have attribute value a_j? Then:
  - $P(a_j|c_i) = 0$, and
  - $P(a_1,\dots,a_n \mid c_i) = \prod_j P(a_j \mid c_i) = 0$
  - no matter what are the values of other attributes
- For example:
  - x = <Sunny, Hot, None, Weak>
  - P(Hot|Yes) = 0, hence
  - P(Yes|x) = 0

Solution

- Let m denote the number of possible values of attribute a_j
- For each class let us consider adding m “virtual examples” with different values of a_j
- Bayesian estimate for $P(a_j|c_i)$ becomes:

$$P(a_j \mid c_i) = \frac{n_{ciaj} + 1}{n_{ci} + m}$$

- Where:
  - n_{ci} – number of training examples with class c_i
  - n_{ciaj} – number of training examples with class c_i and attribute a_j

Learning to classify text

- For example: is an e-mail a spam?
- Represent each document by a set of words
  - Independence assumptions:
    - Order of words does not matter
    - Co-occurrences of words do not matter
- Learning: estimate from training documents:
  - For every class c_i estimate $P(c_i)$
  - For every word w and class c_i estimate $P(w|c_i)$
- Classification: maximum likelihood

Learning in detail

- Vocabulary = all distinct words in training text
- For each class c_i
  - $P(c_i) = \frac{Number\ of\ documents\ of\ class\ c_i}{Total\ number\ of\ documents}$
  - $Text_{ci}$ = concatenated documents of class c_i
  - n_{ci} = total # words in $Text_{ci}$ (count duplicates multiple times)
  - For each word w_j in Vocabulary
    - n_{ciwj} = number of times word w_j occurred in text $Text_{ci}$
    - $P(w_j \mid c_i) = \frac{n_{ciwj} + 1}{n_{ci} + |Vocabulary|}$

Classification in detail

- Index all words in document to classify by j
  - i.e. denote j^{th} word in the document by w_j
- Classify: $c(document) = \arg\max_{c_i} P(c_i) \prod_j P(w_j \mid c_i)$
- In practice $P(w_j|c_i)$ are small so their product is very close to 0; it is better to use:

$$c(document) = \arg\max_{c_i} \log\left[P(c_i) \prod_j P(w_j \mid c_i)\right] =$$

$$= \arg\max_{c_i} \left[\log P(c_i) + \sum_j \log P(w_j \mid c_i)\right]$$

Pre-processing

- Allows adding background knowledge
- May dramatically increase accuracy
- Sample techniques:
  - Lemmatisation - converts words to basic form
  - Stop-list - removes 100 most frequent words

Understanding Naïve Bayes

- Although Naïve Bayes is considered to be subsymbolic, the estimated probabilities may give insight on the classification process
- For example in spam filtering
  - Words with maximum $P(w_j|spam)$ are the words whose presence most predicts en e-mail to be a spam e-mail

Bayesian Belief Networks

- Naïve Bayes assumption of conditional independence of attributes is too restrictive for some problems
- But some assumptions need to be made to allow generalization
- Bayesian Belief Networks assume conditional independence among subset of attributes
- Allows combining prior knowledge about (in)dependencies among attributes

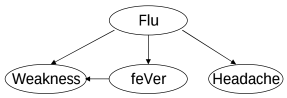
Conditional independence

- X is conditionally independent of Y given Z if
  - $\forall x,y,z$: $P(X=x \mid Y=y, Z=z) = P(X=x \mid Z=z)$
  - Usually written: $P(X|Y,Z) = P(X|Z)$
  - Example: P(Thunder|Rain,Ligthining)=P(Thunder|Lightning)
- Used by Naïve Bayes:
  - $P(A_1,A_2|C) = P(A_1|A_2,C)\ P(A_2|C) = P(A_1|C)P(A_2|C)$

Always true

Only true if A_1 and A_2 conditionally independent

Bayesian Belief Network

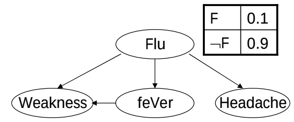


- Connections describe dependence & causality
  - Each node is conditionally independent of its nondescendants, given its immediate predecessors
  - Examples:
    - feVer and Headache are independent given flu
    - feVer and weakness are not independent given flu

Learning Bayesian Network

- Probabilities of attribute values given parents can be estimated from the training set

F	0.1
¬F	0.9



	FV	¬FV	F¬V	¬F ¬V
W	0.9	0.8	0.5	0.05
¬W	0.1	0.2	0.5	0.95

	F	¬F
V	0.8	0.1
¬V	0.2	0.9

	F	¬F
H	0.6	0.1
¬H	0.4	0.9

Inference

- During Bayesian classification we compute:

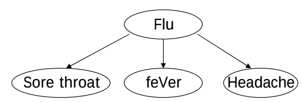
$$c(x) = \arg\max_{c_i} P(a_1,\dots,a_n \mid c_i)P(c_i) = \arg\max_{c_i} P(a_1,\dots,a_n,c_i)$$

- In general in Bayesian network with nodes Y_i:

$$P(y_1,\dots,y_n) = \prod_{i=1}^{n} P(y_i \mid Parents(Y_i))$$

  - Thus $P(a_1,\dots,a_n,c) = \prod_{i=1}^{n} P(a_i \mid Parents(A_i))P(c)$
- Example: Classify patient: W,V,¬H
  - P(W,V,¬H,F) = P(W|VF) P(V|F) P(¬H|F) P(F)

Naïve Bayes network



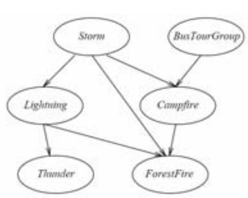
- In case of this network:

$$P(a_1,\dots,a_n,c) = \prod_{i=1}^{n} P(a_i \mid Parents(A_i))P(c)$$

$$= \prod_{i=1}^{n} P(a_i \mid c)P(c)$$

Extensions to Bayesian nets

- Network with hidden states, e.g.



- Learning structure of the network from data

Summary

- Inductive bias of Naïve Bayes:
  - Attributes are independent.
- Although this assumption is often violated, it provides a very efficient tool often used
  - E.g. For spam filtering.
- Applicable to data:
  - with many attributes (possibly missing),
  - which take discrete values (e.g. words).
- Bayesian belief networks
  - Allow prior knowledge about dependencies