Naïve Bayes Classifiers

Simple (naïve) classification methods
based on Bayes rules

What does Naïve mean?

- “Naïve” refers to the (naïve) assumption that data attributes are independent
- The Bayesian method can still be optimal even when this attribute independency is violated (***Domingos, P., and M. Pazzani.*** *1997*)

Properties of Bayes Classifier

- *Combines prior knowledge and observed data:* prior probability of a hypothesis multiplied with probability of the hypothesis given the training data
- *Probabilistic hypothesis:* outputs not only a classification, but a probability distribution over all classes

Bayes classifiers

Assumption: training set consists of instances of different classes described c_j as conjunctions of attributes values

Task: Classify a new instance *d* based on a tuple of attribute values into one of the classes $c_j \in C$

Key idea: assign the most probable class c_{MAP} using Bayes Theorem.

$$
\begin{aligned}
c_{MAP} &= \underset{c_j \in C}{\operatorname{argmax}} P(c_j \mid x_1, x_2, \ldots, x_n) \\
&= \underset{c_j \in C}{\operatorname{argmax}} \frac{P(x_1, x_2, \ldots, x_n \mid c_j) P(c_j)}{P(x_1, x_2, \ldots, x_n)} \\
&= \underset{c_j \in C}{\operatorname{argmax}} P(x_1, x_2, \ldots, x_n \mid c_j) P(c_j)
\end{aligned}
$$

Parameters estimation

- Use the frequencies in the data (MLE)
- $P(c_j)$
  - Can be estimated from the frequency of classes in the training examples.
- $P(x_1,x_2,\ldots,x_n|c_j)$
  - $O(|X|^n \cdot |C|)$ parameters
  - Require large number of training examples
- Independence assumption: attribute values are conditionally independent given the target value: ***naïve*** *Bayes*.

$$P(x_1, x_2, \ldots, x_n \mid c_j) = \prod_i P(x_i \mid c_j)$$

$$c_{NB} = \arg\max_{c_j \in C} P(c_j) \prod_i P(x_i \mid c_j)$$

greatly reduces the number of parameters & data sparseness

Bayes classification

- An unseen instance is classified by computing the class that maximizes the posterior

$$P(x_1, x_2, \ldots, x_n \mid c_j) \qquad P(x_i \mid c_j)$$

- When conditioned independence is satisfied, Naïve Bayes corresponds to MAP classification.

$$P(c_j)$$

$$P(x_i \mid c_j)$$

Example: ‘play tennis’ data

Day	Outlook	Temperature	Humidity	Wind	Play Tennis
Day1	Sunny	Hot	High	Weak	*No*
Day2	Sunny	Hot	High	Strong	*No*
Day3	Overcast	Hot	High	Weak	Yes
Day4	Rain	Mild	High	Weak	Yes
Day5	Rain	Cool	Normal	Weak	Yes
Day6	Rain	Cool	Normal	Strong	*No*
Day7	Overcast	Cool	Normal	Strong	Yes
Day8	Sunny	Mild	High	Weak	*No*
Day9	Sunny	Cool	Normal	Weak	Yes
Day10	Rain	Mild	Normal	Weak	Yes
Day11	Sunny	Mild	Normal	Strong	Yes
Day12	Overcast	Mild	High	Strong	Yes
Day13	Overcast	Hot	Normal	Weak	Yes
Day14	Rain	Mild	High	Strong	*No*

Question: For the day <sunny, cool, high, strong>, what’s the play prediction?

Naive Bayes solution

Classify any new datum instance $\mathbf{x}=(x_1,\ldots x_n)$ *as:*

$$\begin{aligned} C_{NB} &= \arg\max_{c_j \in C} P(c_j) P(x_1, x_2, \cdots, x_n | c_j) \\ &= \arg\max_{c_j \in C} P(c_j) \prod_i^n P(x_i | c_j) \end{aligned}$$

To do this based on training examples, we need to estimate the parameters from the training examples: $P(c_j)$ & $P(x_i|c_j)$

Based on the examples in the table, classify the following datum ***x***:

x=(O=Sunny, T=Cool, H=High, W=strong)

- That means: Play tennis or not?

$$c_{NB} = \arg\max_{c_j} P(c_j) \prod_i P(x_i|c_j)$$
$$= \arg\max_{c_j} P(c_j) P(O = sunny|c_j) P(T = cool|c_j) P(H = high|c_j) P(W = strong|c_j)$$

- Working:

$$P(PlayTennis = yes) = 9/14 = 0.64$$
$$P(PlayTennis = no) = 5/14 = 0.36$$
$$P(Wind = strong \mid PlayTennis = yes) = 3/9 = 0.33$$
$$P(Wind = strong \mid PlayTennis = no) = 3/5 = 0.60$$
$$etc.$$
$$P(yes)P(sunny \mid yes)P(cool \mid yes)P(high \mid yes)P(strong \mid yes) = 0.0053$$
$$P(no)P(sunny \mid no)P(cool \mid no)P(high \mid no)P(strong \mid no) = \mathbf{0.0206}$$
$$\Rightarrow answer : PlayTennis(x) = no$$

Underflow prevention

- Multiplying lots of probabilities, which are between 0 and 1 by definition, can result in floating-point underflow.
- Since *log(xy) = log(x) + log(y),* it is better to perform all computations by summing logs of probabilities rather than multiplying probabilities.
- Class with highest final un-normalized log probability score is still the most probable.

$$c_{NB} = \underset{c_j \in C}{\operatorname{argmax}} \log P(c_j) + \sum_{i \in positions} \log P(x_i \mid c_j)$$

Naïve Bayes for document classification

Relies on a very simple representation of document: Bag of words

$$\gamma\left(\boxed{\text{I like this movie. It is sweet, and makes me laugh. I will definitely recommend it. I'd like to watch it one more time!}}\right) = C$$

C: like-it
dislike-it

Use all the words; or
a subset of the words

Naïve Bayes for document classification

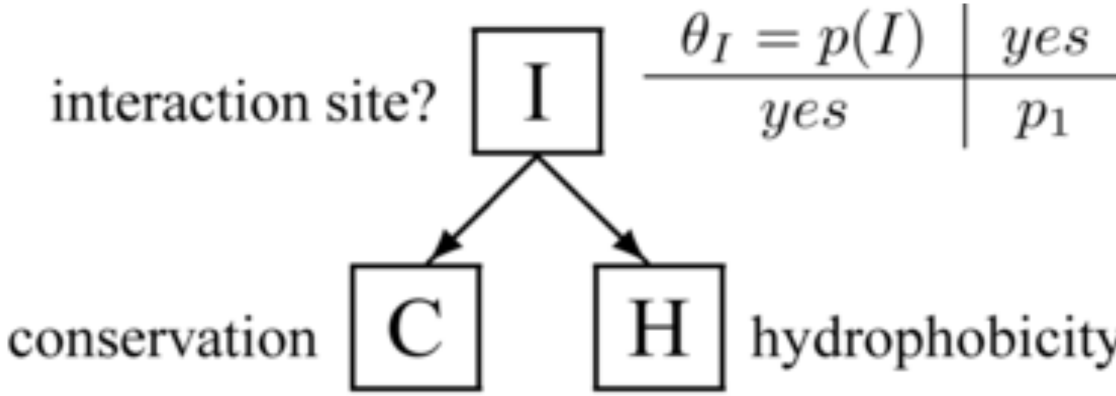
Relies on a very simple representation of document: Bag of words

γ (

word	count
like	2
laugh	1
recommen d	1
..	..
..	..

) = C

Naïve Bayes for interaction site prediction



$\theta_I = p(I)$	yes	no
yes	p_1	$1 - p_1$

$\theta_C = p(C\|I)$	$high$	low
yes	p_2	$1 - p_2$
no	p_3	$1 - p_3$

$\theta_H = p(H\|I)$	$high$	low
yes	p_4	$1 - p_4$
no	p_5	$1 - p_5$

Figure 7. Naïve Bayes Classifier with Model Parameters in the Form of CPTs

Needham CJ, Bradford JR, Bulpitt AJ, Westhead DR (2007) A Primer on Learning in Bayesian Networks for Computational Biology. PLoS Comput Biol 3(8): e129. doi:10.1371/journal.pcbi.0030129
http://www.ploscompbiol.org/article/info:doi/10.1371/journal.pcbi.0030129

Naïve Bayes for Sequence Classification

- RDP classifier – for taxonomic assignment of ribosomal sequences

- A sequence is represented as a bag of k-mers (words)

RDP classifier

- **Naïve Bayesian Classifier for Rapid Assignment of rRNA Sequences into the New Bacterial Taxonomy**
- The Classifier uses a feature space consisting of all possible 8-base subsequences (words).
  - Word sizes between 6 and 9 bases were tested in preliminary experiments: Sizes of 8 and 9 bases gave nearly identical results, while sizes of 6 and 7 bases were less accurate.
- The position of a word in a sequence is ignored. As with text-based Bayesian classifiers, only those words occurring in the query contribute to the score

RDP classifier: training

- Word-specific priors: $W = \{w_1, w_2, \ldots, w_d\}$
  - Given N sequences, let $n(w_i)$ be the number of sequences containing subsequence w_i, $P_i = [n(w_i) + 0.5]/(N + 1)$ (the likelihood of observing word w_i in an rRNA sequence).
- Genus-specific conditional probabilities.
  - For genus G with M sequences, let $m(w_i)$ be the number of these sequences containing word w_i. The conditional probability that a member of G contains $w_{i:}$ $P(w_i|G) = [m(w_i) + P_i]/(M + 1)$.
  - Ignoring the dependency between words in an individual sequence, the joint probability of observing from genus G a (partial) sequence, S, containing a set of words, $V = \{v_1, v_2, \ldots, v_f\}$ ($V \sqsubseteq W$), was estimated as $P(S|G) = \prod P(v_i|G)$.

RDP classifier: prediction

- By Bayes' theorem, the probability that an unknown query sequence, S, is a member of genus G is $P(G|S) = P(S|G) \times P(G)/P(S)$, where $P(G)$ is the prior probability of a sequence being a member of G and $P(S)$ the overall probability of observing sequence S (from any genus).
- Assuming all genera are equally probable (equal priors), the constant terms $P(G)$ and $P(S)$ can be ignored.
- The sequence will be classified as as a member of the genus giving the highest probability score, but we ignore the actual numerical probability estimate.