Naive Bayes Classifier

Along with decision trees, neural networks, nearest nbr, one of the most practical learning methods.

When to use

- Moderate or large training set available
- Attributes that describe instances are conditionally independent given classification

Successful applications:

- Diagnosis
- Classifying text documents

Naive Bayes Classifier

Assume target function $f : X \to V$, where each instance x described by attributes $\langle a_1, a_2 \dots a_n \rangle$. Most probable value of $f(x)$ is:

$$
\begin{aligned}
v_{MAP} &= \operatorname*{argmax}_{v_j \in V} P(v_j | a_1, a_2 \dots a_n) \\
v_{MAP} &= \operatorname*{argmax}_{v_j \in V} \frac{P(a_1, a_2 \dots a_n | v_j) P(v_j)}{P(a_1, a_2 \dots a_n)} \\
&= \operatorname*{argmax}_{v_j \in V} P(a_1, a_2 \dots a_n | v_j) P(v_j)
\end{aligned}
$$

Naive Bayes assumption:

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

which gives

Naive Bayes classifier: $v_{NB} = \operatorname*{argmax}_{v_j \in V} P(v_j) \prod_i P(a_i | v_j)$

Naive Bayes Algorithm

Naive_Bayes_Learn($examples$)

For each target value v_j

$\hat{P}(v_j) \leftarrow$ estimate $P(v_j)$

For each attribute value a_i of each attribute a

$\hat{P}(a_i|v_j) \leftarrow$ estimate $P(a_i|v_j)$

Classify_New_Instance(x)

$$v_{NB} = \underset{v_j \in V}{\operatorname{argmax}} \hat{P}(v_j) \prod_{a_i \in x} \hat{P}(a_i|v_j)$$

Naive Bayes: Example

Consider *PlayTennis* again, and new instance

$\langle Outlk = sun, Temp = cool, Humid = high, Wind = strong \rangle$

Want to compute:

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

$P(y)\ P(sun|y)\ P(cool|y)\ P(high|y)\ P(strong|y) = .005$

$P(n)\ P(sun|n)\ P(cool|n)\ P(high|n)\ P(strong|n) = .021$

$$\rightarrow v_{NB} = n$$

Naive Bayes: Subtleties

---

1. Conditional independence assumption is often violated

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

- ...but it works surprisingly well anyway. Note don't need estimated posteriors $\hat{P}(v_j|x)$ to be correct; need only that

$$\operatorname*{argmax}_{v_j \in V} \hat{P}(v_j) \prod_i \hat{P}(a_i|v_j) = \operatorname*{argmax}_{v_j \in V} P(v_j) \prod_i P(a_i|v_j)$$

- see [Domingos & Pazzani, 1996] for analysis
- Naive Bayes posteriors often unrealistically close to 1 or 0

Naive Bayes: Subtleties

---

2. what if none of the training instances with target value v_j have attribute value a_i? Then

$$\hat{P}(a_i|v_j) = 0, \text{ and...}$$

$$\hat{P}(v_j) \prod_i \hat{P}(a_i|v_j) = 0$$

Typical solution is Bayesian estimate for $\hat{P}(a_i|v_j)$

$$\hat{P}(a_i|v_j) \leftarrow \frac{n_c + mp}{n + m}$$

where

- n is number of training examples for which $v = v_j$,
- n_c number of examples for which $v = v_j$ and $a = a_i$
- p is prior estimate for $\hat{P}(a_i|v_j)$
- m is weight given to prior (i.e. number of "virtual" examples)

Learning to Classify Text

Why?

- Learn which news articles are of interest
- Learn to classify web pages by topic

Naive Bayes is among most effective algorithms

What attributes shall we use to represent text documents??

Learning to Classify Text

Target concept $Interesting? : Document \rightarrow \{+, -\}$

1. Represent each document by vector of words
   - one attribute per word position in document
2. Learning: Use training examples to estimate
   - $P(+)$
   - $P(-)$
   - $P(doc|+)$
   - $P(doc|-)$

Naive Bayes conditional independence assumption

$$P(doc|v_j) = \prod_{i=1}^{length(doc)} P(a_i = w_k|v_j)$$

where $P(a_i = w_k|v_j)$ is probability that word in position i is w_k, given v_j

one more assumption:
$P(a_i = w_k|v_j) = P(a_m = w_k|v_j), \forall i, m$

LEARN_NAIVE_BAYES_TEXT$(Examples, V)$

1. collect all words and other tokens that occur in $Examples$

- $Vocabulary \leftarrow$ all distinct words and other tokens in $Examples$

2. calculate the required $P(v_j)$ and $P(w_k|v_j)$ probability terms

- For each target value v_j in V do
  - $docs_j \leftarrow$ subset of $Examples$ for which the target value is v_j
  - $P(v_j) \leftarrow \frac{|docs_j|}{|Examples|}$
  - $Text_j \leftarrow$ a single document created by concatenating all members of $docs_j$
  - $n \leftarrow$ total number of words in $Text_j$ (counting duplicate words multiple times)
  - for each word w_k in $Vocabulary$
    * $n_k \leftarrow$ number of times word w_k occurs in $Text_j$
    * $P(w_k|v_j) \leftarrow \frac{n_k+1}{n+|Vocabulary|}$

Classify_naive_Bayes_text(Doc)

- $positions \leftarrow$ all word positions in Doc that contain tokens found in $Vocabulary$
- Return v_{NB}, where

$$v_{NB} = \operatorname*{argmax}_{v_j \in V} P(v_j) \prod_{i \in positions} P(a_i|v_j)$$

Classify Web Pages

Training set: approximately 5000 web pages from 4 university CS sites

	Actual						
Predicted	Course	Student	Faculty	Staff	Research.Project	Department	Other
Course	**202**	17	0	0	1	0	552
Student	0	**421**	14	17	2	0	519
Faculty	5	56	**118**	16	3	0	264
Staff	0	15	1	**4**	0	0	45
Research.Project	8	9	10	5	**62**	0	384
Department	10	8	3	1	5	**4**	209
Other	19	32	7	3	12	0	**1064**

Classify Web Pages

Words ranked by $\log \frac{Pr(w|v_j)}{P(w|\neq v_j)}$

Student

my	0.0247
page	0.0109
home	0.0104
am	0.0085
university	0.0061
computer	0.0060
science	0.0059
me	0.0058
at	0.0049
here	0.0046

Faculty

DDDD	0.0138
of	0.0113
and	0.0109
professor	0.0088
computer	0.0073
research	0.0060
science	0.0057
university	0.0049
DDD	0.0042
systems	0.0042

Staff

rice	0.0023
scout	0.0018
my	0.0016
columbia	0.0015
sekine	0.0013
satoshi	0.0012
liebrock	0.0009
watanabe	0.0007
saskatchewan	0.0007
me	0.0007

Course

course	0.0151
DD:DD	0.0130
homework	0.0106
will	0.0088
D	0.0080
assignments	0.0079
class	0.0073
hours	0.0059
assignment	0.0058
due	0.0058

Research.Project

group	0.0060
project	0.0049
research	0.0049
of	0.0030
laboratory	0.0029
systems	0.0028
and	0.0027
our	0.0026
system	0.0024
projects	0.0020

Department

department	0.0179
science	0.0153
computer	0.0111
faculty	0.0070
information	0.0069
undergraduate	0.0058
graduate	0.0047
staff	0.0045
server	0.0042
courses	0.0042

Other

D	0.0374
DD	0.0246
the	0.0153
eros	0.0010
hplay*D*	0.0097
u*DD*b	0.0067
to	0.0064
bluto	0.0052
gt	0.0050
that	0.0043

Article from rec.sport.hockey

```
Path: cantaloupe.srv.cs.cmu.edu!das-news.harvard.edu!ogicse!uwm.edu!zaphod.mps.ohio-state.edu!usc!aludra.usc.edu!not-for-mail
From: xxx@yyy.zzz.edu (John Doe)
Subject: Re: This year's biggest and worst (opinion)...
Date: 5 Apr 93 09:53:39 GMT

I can only comment on the Kings, but the most
obvious candidate for pleasant surprise is Alex
Zhitnik. He came highly touted as a defensive
defenseman, but he's clearly much more than that.
Great skater and hard shot (though wish he were
more accurate). In fact, he pretty much allowed
the Kings to trade away that huge defensive
liability Paul Coffey. Kelly Hrudey is only the
biggest disappointment if you thought he was any
good to begin with. But, at best, he's only a
mediocre goaltender. A better choice would be
Tomas Sandstrom, though not through any fault of
his own, but because some thugs in Toronto decided
```

to threaten his career in order to avoid conceding a goal. Other than that, the award goes to Robert Lang, an uninspiring Czech. Robitaille could easily be MVP, but I'd prefer to give it to ...

Classify Newsgroup Articles

Classify articles from one of twenty newsgroups

comp.graphics	misc.forsale
comp.os.ms-windows.misc	rec.autos
comp.sys.ibm.pc.hardware	rec.motorcycles
comp.sys.mac.hardware	rec.sport.baseball
comp.windows.x	rec.sport.hockey
soc.religion.christian	sci.space
talk.politics.guns	sci.crypt
talk.politics.mideast	sci.electronics
talk.politics.misc	sci.med
talk.religion.misc	
alt.atheism	

Naive Bayes classifier $\rightarrow$ 89% accuracy

Social Learning

[Shardanand and Maes 1995]

Idea: Learn by correlating user ratings

	d_1	d_2	d_3	d_4	d_5	d_6
User A	2	-	4	4	-	5
User B	1	5	4	-	3	4
User C	5	2	-	2	1	-
User D	3	-	2	2	-	4

Predict

$$rating(user_i, d_n) = \frac{\Sigma_{i \neq j}\, w_{ij} \cdot rating(user_j, d_n)}{\Sigma_{i \neq j}\, w_{ij}}$$

where w_{ij} is a measure of correlation of $user_i$ and $user_j$ interests

Social Learning, Cont'd.

Base algorithm: (i.e., mean)

$$w_{ij} = \frac{1}{\text{number of users}}$$

Mean sq difference:

$$w_{ij} = \frac{1}{\Sigma_d (r(U_i, d) - r(U_j, d))^2}$$

Pearson r correlation:

$$w_{ij} = \frac{\Sigma_d (r(U_i, d) - \overline{U_i})(r(U_j, d) - \overline{U_j})}{\sqrt{\Sigma_d (r(U_i, d) - \overline{U_i})^2 (r(U_j, d) - \overline{U_j})^2}}$$

Constrained *Pearson* r correlation ($4 =$ neutral):

$$w_{ij} = \frac{\Sigma_d (r(U_i, d) - 4)(r(U_j, d) - 4)}{\sqrt{\Sigma_d (r(U_i, d) - 4)^2 (r(U_j, d) - 4)^2}}$$

Results of Social Learning in Ringo

2000 users, 125 musical artists, ratings 1–7

Thresholded sum, considering only terms where $w_{ij} > threshold$

- Base alg: mean error = 1.3, $\sigma = 1.6$
- Constrained Peason r: mean error = 1.1, $\sigma = 1.3$, 97% of cases predictable