ICS Summer Academy Session II
Topic 5: Naive Bayes Classifiers

Michael Shindler

Bayes' Theorem: A Recap

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$

- $P(h)$ is the *prior* probability

- $P(D)$ See supplemental handout.

- $P(D|h)$ probability of the data, given that the hypothesis holds

- But we are more interested in $P(h|D)$, the *posterior* probability

Finding the most probable hypothesis

- We often have a set of candidate hypotheses H
- Goal: which $h \in H$ is most probable given observations D
- This is the *maximum a posteriori* hypothesis:

$$
\begin{aligned}
h_{MAP} &\equiv \arg\max_{h \in H} P(h|D) \\
&= \arg\max_{h \in H} \frac{P(D|h)P(h)}{P(D)} \\
&= \arg\max_{h \in H} P(D|h)P(h)
\end{aligned}
$$

If all h equally likely

$$= \underset{h \in H}{\arg\max}\ P(D|h)$$

Brute Force MAP Learning

- Applying Bayes Theorem to Concept Learning
- For each hypothesis $h \in H$, calculate $P(h|D)$
- Output the hypothesis h_{MAP} with highest $P(h|D)$

- We are going to make three assumptions here:
  1. The training data D is noise free
  2. $c \in H$
  3. No a priori reason to prefer any given hypothesis.

$1/|H|$

- We need to decide values for $P(h)$, $P(D|h)$, and $P(D)$.

Consistent Learners

An algorithm is a **consistent learner** if it always outputs a hypothesis that commits zero training error.

- Name an algorithm we have seen that is a consistent learner.

ID3, Candidate-elim

- Name one we have seen that is not.

Every consistent learner outputs a MAP if:

- Uniform probability distribution over H
- No noise in training data.

Towards a Bayes Optimal Classifier

We were asking: most probable *hypothesis*?

We should ask: most probable *classification* of new instance?

Are these the same thing?

$P(yes) = \sum_i P(yes|h_i) \cdot P(h_i)$

$P(h_1) = .4$

$P(h_2) = .3$

$P(h_3) = .2$

$P(h_4) = .1$

Most probable classification

$P(v_j|D)$: probability correct classification for new instance is v_j:

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

Bayes Optimal Classification:

$$\arg\max_{v_j \in V} \sum_{h_i \in H} P(v_j|h_j)P(h_i|D)$$

Gibbs Algorithm

One downside of Bayes Optimal Classifier is it is expensive.

So here's a less optimal algorithm:

1. Choose $h \in H$ at random $\propto$ posterior
2. Use h to classify next instance

It can be shown that this error rate is at most twice optimal.

Naive Bayes Classifier

We have instances described by many attributes.

$$
\begin{aligned}
v_{MAP} &= \arg\max_{v_j \in V} P(v_j | a_1, a_2, \ldots, a_n) \\
&= \arg\max_{v_j \in V} \frac{P(a_1, a_2, \ldots, a_n | v_j) P(v_j)}{P(a_1, a_2, \ldots a_n)} \\
&= \arg\max_{v_j \in V} P(a_1, a_2, \ldots, a_n | v_j) P(v_j)
\end{aligned}
$$

We assume all attributes are *conditionally independent* given target value

PlayTennis example revisited

Outlook	Temperature	Humidity	Wind	PlayTennis
Sunny	Hot	High	Weak	No
Sunny	Hot	High	Strong	No
Overcast	Hot	High	Weak	Yes
Rain	Mild	High	Weak	Yes
Rain	Cool	Normal	Weak	Yes
Rain	Cool	Normal	Strong	No
Overcast	Cool	Normal	Strong	Yes
Sunny	Mild	High	Weak	No
Sunny	Cool	Normal	Weak	Yes
Rain	Mild	Normal	Weak	Yes
Sunny	Mild	Normal	Strong	Yes
Overcast	Mild	High	Strong	Yes
Overcast	Hot	Normal	Weak	Yes
Rain	Mild	High	Strong	No

Estimate $P(\text{PlayTennis} = \text{Yes})$ and $P(\text{PlayTennis} = \text{No})$

Prediction of values

9/14 yes

5/14 no

PlayTennis example revisited

Outlook	Temperature	Humidity	Wind	PlayTennis
Sunny	Hot	High	Weak	No
Sunny	Hot	High	Strong	No
Overcast	Hot	High	Weak	Yes
Rain	Mild	High	Weak	Yes
Rain	Cool	Normal	Weak	Yes
Rain	Cool	Normal	Strong	No
Overcast	Cool	Normal	Strong	Yes
Sunny	Mild	High	Weak	No
Sunny	Cool	Normal	Weak	Yes
Rain	Mild	Normal	Weak	Yes
Sunny	Mild	Normal	Strong	Yes
Overcast	Mild	High	Strong	Yes
Overcast	Hot	Normal	Weak	Yes
Rain	Mild	High	Strong	No

Estimate $P(\text{Wind=strong}|\text{PlayTennis} = \text{Yes})$ and similarly for no.

3/9

3/5

PlayTennis example revisited

(This is an estimator)

Outlook	Temperature	Humidity	Wind	PlayTennis
Sunny	Hot	High	Weak	No
Sunny	Hot	High	Strong	No
Overcast	Hot	High	Weak	Yes
Rain	Mild	High	Weak	Yes
Rain	Cool	Normal	Weak	Yes
Rain	Cool	Normal	Strong	No
Overcast	Cool	Normal	Strong	Yes
Sunny	Mild	High	Weak	No
Sunny	Cool	Normal	Weak	Yes
Rain	Mild	Normal	Weak	Yes
Sunny	Mild	Normal	Strong	Yes
Overcast	Mild	High	Strong	Yes
Overcast	Hot	Normal	Weak	Yes
Rain	Mild	High	Strong	No

Do we play tennis in this circumstance?

$x =$

Outlook	Temperature	Humidity	Wind
Sunny	cool	high	strong

$\cdot P(\text{temp}=\text{cool} \mid \text{yes})$

$P(\text{yes} \mid x) = P(\text{yes}) \cdot P(\text{Sunny} \mid \text{yes}) \cdot P(\text{high} \cdot \text{yes})\, P(\text{str} \mid \text{yes})$

Estimating Probabilities

In Naive Bayes, we estimated $P(x|y)$ as

$$\frac{\#\text{ observed } x \text{ and } y \text{ together } + mp}{\#\text{ observed } y + m}$$

When might this be a poor idea?

p = prior est.

m = equiv sample size

Training Phase: Find the Spammer's Dictionary

During training phase, we have lots of emails.

$P(\text{spam}) = \%$ labeled as spam

For each word w_i,
$P(w_i|\text{spam}) = \%$ of words in spam that are w_i

$P(x|\text{spam}) = \prod_i P(w_i|\text{spam})^{\# w_i}$

Product (similar to Σ but multiply)

This leads to a very simple algorithm

Training Data: Large number of emails, labeled spam or not.

End result: given an email:

- Count the words
- Apply weights of spam, not-spam categories
- Make a decision: if spam score is higher, it's spam.

$P(\text{spam} \mid x)$ $\qquad$ $P(\text{spam} \mid \neg x)$

Classifier in the linear form of compatibility scores

$$
\begin{aligned}
\log[P(x|\mathsf{spam})P(\mathsf{spam})] &= \log\left[\prod_i P(w_i|\mathsf{spam})^{\#w_i} P(\mathsf{spam})\right] \\
&= \sum_i \#w_i \log P(w_i|\mathsf{spam}) + \log P(\mathsf{spam})
\end{aligned}
$$

Concluding Remarks

- Naive Bayes are probabilistic classification models

- Naive Bayes are **generative** classification models

Concluding Remarks

- Naive Bayes assumption helps in high-dimensional settings

- Naive Bayes are robust to isolated noise points

Concluding Remarks

- Naive Bayes can handle missing values in a training set

- Naive Bayes are robust to irrelevant attributes