Naive Bayes Classifiers

Connectionist and Statistical Language Processing

Frank Keller

keller@coli.uni-sb.de

Computerlinguistik

Universität des Saarlandes

Overview

- Sample data set with frequencies and probabilities
- Classification based on Bayes rule
- Maximum a posterior and maximum likelihood
- Properties of Bayes classifiers
- Naive Bayes classifiers
- Parameter estimation, properties, example
- Dealing with sparse data
- Application: email classification

Literature: Witten and Frank (2000: ch. 4), Mitchell (1997: ch. 6).

A Sample Data Set

Fictional data set that describes the weather conditions for playing some unspecified game.

outlook	temp.	humidity	windy	play
sunny	hot	high	false	no
sunny	hot	high	true	no
overcast	hot	high	false	yes
rainy	mild	high	false	yes
rainy	cool	normal	false	yes
rainy	cool	normal	true	no
overcast	cool	normal	true	yes

outlook	temp.	humidity	windy	play
sunny	mild	high	false	no
sunny	cool	normal	false	yes
rainy	mild	normal	false	yes
sunny	mild	normal	true	yes
overcast	mild	high	true	yes
overcast	hot	normal	false	yes
rainy	mild	high	true	no

Frequencies and Probabilities

Frequencies and probabilities for the weather data:

outlook			temperature			humidity			windy			play	
	yes	no		yes	no		yes	no		yes	no	yes	no
sunny	2	3	hot	2	2	high	3	4	false	6	2	9	5
overcast	4	0	mild	4	2	normal	6	1	true	3	3		
rainy	3	2	cool	3	1								
	yes	no		yes	no		yes	no		yes	no	yes	no
sunny	2/9	3/5	hot	2/9	2/5	high	3/9	4/5	false	6/9	2/5	9/14	5/14
overcast	4/9	0/5	mild	4/9	2/5	normal	6/9	1/5	true	3/9	3/5		
rainy	3/9	2/5	cool	3/9	1/5								

Classifying an Unseen Example

Now assume that we have to classify the following new instance:

outlook	temp.	humidity	windy	play
sunny	cool	high	true	?

Key idea: compute a probability for each class based on the probability distribution in the training data.

First take into account the the probability of each attribute. Treat all attributes **equally important**, i.e., multiply the probabilities:

$$P(\text{yes}) = 2/9 \cdot 3/9 \cdot 3/9 \cdot 3/9 = 0.0082$$
$$P(\text{no}) = 3/5 \cdot 1/5 \cdot 4/5 \cdot 3/5 = 0.0577$$

Classifying an Unseen Example

Now take into account the **overall probability** of a given class. Multiply it with the probabilities of the attributes:

$$P(\text{yes}) = 0.0082 \cdot 9/14 = 0.0053$$
$$P(\text{no}) = 0.0577 \cdot 5/14 = 0.0206$$

Now choose the class so that it **maximizes** this probability. This means that the new instance will be classified as no.

Bayes Rule

This procedure is based on *Bayes Rule*, which says: if you have a hypothesis h and data D which bears on the hypothesis, then:

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)} \tag{1}$$

$P(h)$: independent probability of h: *prior probability*
$P(D)$: independent probability of D
$P(D|h)$: conditional probability of D given h: *likelihood*
$P(h|D)$: cond. probability of h given D: *posterior probability*

Maximum A Posteriori

Based on Bayes Rule, we can compute the *maximum a posteriori* hypothesis for the data:

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

H: set of all hypotheses

Note that we can drop $P(D)$ as the probability of the data is constant (and independent of the hypothesis).

Maximum Likelihood

Now assume that all hypotheses are equally probable a priori, i.e, $P(h_i) = P(h_j)$ for all $h_i, h_j \in H$.

This is called assuming a *uniform prior*. It simplifies computing the posterior:

$$h_{\mathrm{ML}} = \arg\max_{h \in H} P(D|h) \tag{3}$$

This hypothesis is called the *maximum likelihood hypothesis*.

Properties of Bayes Classifiers

- **Incrementality:** with each training example, the prior and the likelihood can be updated dynamically: flexible and robust to errors.
- **Combines prior knowledge and observed data:** prior probability of a hypothesis multiplied with probability of the hypothesis given the training data.
- **Probabilistic hypotheses:** outputs not only a classification, but a probability distribution over all classes.
- **Meta-classification:** the outputs of several classifiers can be combined, e.g., by multiplying the probabilities that all classifiers predict for a given class.

Naive Bayes Classifier

Assumption: training set consists of instances described as **conjunctions of attributes values**, target classification based on finite set of classes V.

The task of the learner is to predict the correct class for a new instance $\langle a_1, a_2, \ldots, a_n \rangle$.

Key idea: assign most probable class v_{MAP} using Bayes Rule.

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

Naive Bayes: Parameter Estimation

Estimating $P(v_j)$ is simple: compute the relative frequency of each target class in the training set.

Estimating $P(a_1, a_2, \ldots, a_n | v_j)$ is difficult: typically not enough instances for each attribute combination in the training set: *sparse data problem.*

Independence assumption: attribute values are conditionally independent given the target value: **naive** Bayes.

$$P(a_1, a_2, \ldots, a_n | v_j) = \prod_i P(a_i | v_j) \tag{5}$$

Hence we get the following classifier:

$$v_{\mathrm{NB}} = \arg\max_{v_j \in V} P(v_j) \prod_i P(a_i | v_j) \tag{6}$$

Naive Bayes: Properties

- Estimating $P(a_i|v_j)$ instead of $P(a_1, a_2, \ldots, a_n|v_j)$ greatly reduces the number of parameters (and data sparseness).
- The learning step in Naive Bayes consists of estimating $P(a_i|v_j)$ and $P(v_j)$ based on the frequencies in the training data.
- There is no explicit search during training (as opposed to decision trees).
- An unseen instance is classified by computing the class that maximizes the posterior.
- When conditional independence is satisfied, Naive Bayes corresponds to MAP classification.

Naive Bayes: Example

Apply Naive Bayes to the weather training data. The hypothesis space is $V = \{\text{yes}, \text{no}\}$. Classify the following new instance:

outlook	temp.	humidity	windy	play
sunny	cool	high	true	?

$$
\begin{aligned}
v_{\text{NB}} &= \arg \max_{v_j \in \{\text{yes},\text{no}\}} P(v_j) \prod_i P(a_i | v_j) \\
&= \arg \max_{v_j \in \{\text{yes},\text{no}\}} P(v_j) P(\text{outlook} = \text{sunny} | v_j) P(\text{temp} = \text{cool} | v_j) \\
&\quad P(\text{humidity} = \text{high} | v_j) P(\text{windy} = \text{true} | v_j)
\end{aligned}
$$

Compute priors:

$$P(\text{play} = \text{yes}) = 9/14 \quad P(\text{play} = \text{no}) = 5/14$$

Naive Bayes: Example

Compute conditionals (examples):

$$P(\text{windy} = \text{true}|\text{play} = \text{yes}) = 3/9$$

$$P(\text{windy} = \text{true}|\text{play} = \text{no}) = 3/5$$

Then compute the best class:

$$P(\text{yes})P(\text{sunny}|\text{yes})P(\text{cool}|\text{yes})P(\text{high}|\text{yes})P(\text{true}|\text{yes})$$

$$= 9/14 \cdot 2/9 \cdot 3/9 \cdot 3/9 \cdot 3/9 = 0.0053$$

$$P(\text{no})P(\text{sunny}|\text{no})P(\text{cool}|\text{no})P(\text{high}|\text{no})P(\text{true}|\text{no})$$

$$= 5/14 \cdot 3/5 \cdot 1/5 \cdot 4/5 \cdot 3/5 = 0.0206$$

Now classify the unseen instance:

$$v_{\text{NB}} = \arg \max_{v_j \in \{\text{yes,no}\}} P(v_j)P(\text{sunny}|v_j)P(\text{cool}|v_j)P(\text{high}|v_j)P(\text{true}|v_j)$$

$$= \text{no}$$

Naive Bayes: Sparse Data

Conditional probabilities can be estimated directly as relative frequencies:

$$P(a_i|v_j) = \frac{n_c}{n}$$

where n is the total number of training instances with class v_j, and n_c is the number of instances with attribute a_i and class v_i.

Problem: this provides a poor estimate if n_c is very small.

Extreme case: if $n_c = 0$, then the whole posterior will be zero.

Naive Bayes: Sparse Data

Solution: use the m-**estimate** of probabilities:

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

p: prior estimate of the probability
m: equivalent sample size (constant)

In the absence of other information, assume a **uniform prior:**

$p = \frac{1}{k}$

where k is the number of values that the attribute a_i can take.

Application: Email Classification

Training data: a corpus of email messages, each message annotated as spam or no spam.

Task: classify new email messages as spam/no spam.

To use a naive Bayes classifier for this task, we have to first find an *attribute representation* of the data.

Treat each text position as an attribute, with as its value the word at this position. Example: email starts: *get rich*.

The naive Bayes classifier is then:

$$\begin{aligned} v_{\text{NB}} &= \arg \max_{v_j \in \{\text{spam,nospam}\}} P(v_j) \prod_i P(a_i|v_j) \\ &= \arg \max_{v_j \in \{\text{spam,nospam}\}} P(v_j) P(a_1 = get|v_j) P(a_2 = rich|v_j) \end{aligned}$$

Application: Email Classification

Using naive Bayes means we assume that **words are independent of each** other. Clearly incorrect, but doesn't hurt a lot for our task.

The classifier uses $P(a_i = w_k | v_j)$, i.e., the probability that the i-th word in the email is the k-word in our vocabulary, given the email has been classified as v_j.

Simplify by assuming that **position is irrelevant**: estimate $P(w_k | v_j)$, i.e., the probability that word w_k occurs in the email, given class v_j.

Create a **vocabulary:** make a list of all words in the training corpus, discard words with very high or very low frequency.

Application: Email Classification

Training: estimate priors:

$P(v_j) = \frac{n}{N}$

Estimate likelihoods using the **m-estimate:**

$P(w_k|v_j) = \frac{n_k+1}{n+|Vocabulary|}$

N: total number of words in all emails
n: number of words in emails with class v_j
n_k: number of times word w_k occurs in emails with class v_j
$|Vocabulary|$: size of the vocabulary

Testing: to classify a new email, assign it the class with the highest posterior probability. Ignore unknown words.

Summary

- Bayes classifier combines prior knowledge with observed data: assigns a posterior probability to a class based on its prior probability and its likelihood given the training data.
- Computes the maximum a posterior (MAP) hypothesis or the maximum likelihood (ML) hypothesis.
- Naive Bayes classifier assumes conditional independence between attributes and assigns the MAP class to new instances.
- Likelihoods can be estimated based on frequencies. Problem: sparse data. Solution: using the m-estimate (adding a constant).

References

Mitchell, Tom. M. 1997. *Machine Learning*. New York: McGraw-Hill.

Witten, Ian H., and Eibe Frank. 2000. *Data Mining: Practical Machine Learing Tools and Techniques with Java Implementations*. San Diego, CA: Morgan Kaufmann.