The Naïve Bayes Classifier

Machine Learning

THE UNIVERSITY OF UTAH

Today's lecture

- The naïve Bayes Classifier
- Learning the naïve Bayes Classifier
- Practical concerns

Today's lecture

- The naïve Bayes Classifier
- Learning the naïve Bayes Classifier
- Practical concerns

Where are we?

We have seen Bayesian learning

- Using a probabilistic criterion to select a hypothesis
- Maximum a posteriori and maximum likelihood learning

  You should know what is the difference between them

Where are we?

We have seen Bayesian learning

- Using a probabilistic criterion to select a hypothesis
- Maximum a posteriori and maximum likelihood learning

  You should know what is the difference between them

We could also learn functions that *predict* probabilities of outcomes

- Different from using a probabilistic criterion to learn

Maximum a posteriori (MAP) prediction as opposed to MAP learning

MAP prediction

Using the Bayes rule for predicting y given an input $\mathbf{x}$

$$P(Y = y \mid X = \mathbf{x}) = \frac{P(X = \mathbf{x} \mid Y = y)P(Y = y)}{P(X = \mathbf{x})}$$

Posterior probability of label being y for this input $\mathbf{x}$

MAP prediction

Using the Bayes rule for predicting y given an input $\mathbf{x}$

$$P(Y = y \mid X = \mathbf{x}) = \frac{P(X = \mathbf{x} \mid Y = y)P(Y = y)}{P(X = \mathbf{x})}$$

Predict the label y for the input $\mathbf{x}$ using

$$\operatorname*{argmax}_y \frac{P(X = \mathbf{x} \mid Y = y)P(Y = y)}{P(X = \mathbf{x})}$$

MAP prediction

Using the Bayes rule for predicting y given an input $\mathbf{x}$

$$P(Y = y \mid X = \mathbf{x}) = \frac{P(X = \mathbf{x} \mid Y = y)P(Y = y)}{P(X = \mathbf{x})}$$

Predict the label y for the input $\mathbf{x}$ using

$$\underset{y}{\operatorname{argmax}}\, P(X = \mathbf{x} \mid Y = y)P(Y = y)$$

MAP prediction

Don't confuse with *MAP learning*: finds hypothesis by

$$h_{MAP} = \arg\max_{h \in H} P(D|h)P(h)$$

Using the Bayes rule for predicting y given an input $\mathbf{x}$

$$P(Y = y \mid X = \mathbf{x}) = \frac{P(X = \mathbf{x} \mid Y = y)P(Y = y)}{P(X = \mathbf{x})}$$

Predict the label y for the input $\mathbf{x}$ using

$$\underset{y}{\operatorname{argmax}} P(X = \mathbf{x} \mid Y = y)P(Y = y)$$

MAP prediction

Predict the label y for the input $\mathbf{x}$ using

$$\operatorname*{argmax}_{y} \boxed{P(X = \mathbf{x} \mid Y = y)}\boxed{P(Y = y)}$$

Likelihood of observing this input $\mathbf{x}$ when the label is y

Prior probability of the label being y

All we need are these two sets of probabilities

Example: Tennis again

Example: Tennis again

	Play tennis	P(Play tennis)
Prior	Yes	0.3
	No	0.7

Without any other information, what is the prior probability that I should play tennis?

Example: Tennis again

Prior

Play tennis	P(Play tennis)
Yes	0.3
No	0.7

Without any other information, what is the prior probability that I should play tennis?

Likelihood

Temperature	Wind	P(T, W |Tennis = Yes)
Hot	Strong	0.15
Hot	Weak	0.4
Cold	Strong	0.1
Cold	Weak	0.35

On days that I do play tennis, what is the probability that the temperature is T and the wind is W?

Temperature	Wind	P(T, W |Tennis = No)
Hot	Strong	0.4
Hot	Weak	0.1
Cold	Strong	0.3
Cold	Weak	0.2

On days that I don't play tennis, what is the probability that the temperature is T and the wind is W?

Example: Tennis again

Prior

Play tennis	P(Play tennis)
Yes	0.3
No	0.7

Likelihood

Temperature	Wind	P(T, W |Tennis = Yes)
Hot	Strong	0.15
Hot	Weak	0.4
Cold	Strong	0.1
Cold	Weak	0.35

Temperature	Wind	P(T, W |Tennis = No)
Hot	Strong	0.4
Hot	Weak	0.1
Cold	Strong	0.3
Cold	Weak	0.2

Input:
Temperature = Hot (H)
Wind = Weak (W)

Should I play tennis?

Example: Tennis again

Prior

Play tennis	P(Play tennis)
Yes	0.3
No	0.7

Likelihood

Temperature	Wind	P(T, W |Tennis = Yes)
Hot	Strong	0.15
Hot	Weak	0.4
Cold	Strong	0.1
Cold	Weak	0.35

Temperature	Wind	P(T, W |Tennis = No)
Hot	Strong	0.4
Hot	Weak	0.1
Cold	Strong	0.3
Cold	Weak	0.2

Input:
Temperature = Hot (H)
Wind = Weak (W)

Should I play tennis?

$\text{argmax}_y\ P(H, W \mid \text{play?})\ P(\text{play?})$

Example: Tennis again

Input:
Temperature = Hot (H)
Wind = Weak (W)

Should I play tennis?

Prior

Play tennis	P(Play tennis)
Yes	0.3
No	0.7

Likelihood

Temperature	Wind	P(T, W | Tennis = Yes)
Hot	Strong	0.15
Hot	Weak	0.4
Cold	Strong	0.1
Cold	Weak	0.35

Temperature	Wind	P(T, W | Tennis = No)
Hot	Strong	0.4
Hot	Weak	0.1
Cold	Strong	0.3
Cold	Weak	0.2

$\text{argmax}_y P(H, W \mid play?)\, P(play?)$

$P(H, W \mid Yes)\, P(Yes) = 0.4 \times 0.3 = 0.12$

$P(H, W \mid No)\, P(No) = 0.1 \times 0.7 = 0.07$

Example: Tennis again

Prior

Play tennis	P(Play tennis)
Yes	0.3
No	0.7

Likelihood

Temperature	Wind	P(T, W | Tennis = Yes)
Hot	Strong	0.15
Hot	Weak	0.4
Cold	Strong	0.1
Cold	Weak	0.35

Temperature	Wind	P(T, W | Tennis = No)
Hot	Strong	0.4
Hot	Weak	0.1
Cold	Strong	0.3
Cold	Weak	0.2

Input:
Temperature = Hot (H)
Wind = Weak (W)

Should I play tennis?

$\text{argmax}_y\ P(H, W \mid play?)\ P(play?)$

$P(H, W \mid Yes)\ P(Yes) = 0.4 \times 0.3$
$= 0.12$

$P(H, W \mid No)\ P(No) = 0.1 \times 0.7$
$= 0.07$

MAP prediction = Yes

How hard is it to learn probabilistic models?

	O	T	H	W	Play?
1	S	H	H	W	-
2	S	H	H	S	-
3	O	H	H	W	+
4	R	M	H	W	+
5	R	C	N	W	+
6	R	C	N	S	-
7	O	C	N	S	+
8	S	M	H	W	-
9	S	C	N	W	+
10	R	M	N	W	+
11	S	M	N	S	+
12	O	M	H	S	+
13	O	H	N	W	+
14	R	M	H	S	-

Outlook: S(unny), O(vercast), R(ainy)

Temperature: H(ot), M(edium), C(ool)

Humidity: H(igh), N(ormal), L(ow)

Wind: S(trong), W(eak)

How hard is it to learn probabilistic models?

	O	T	H	W	Play?
1	S	H	H	W	-
2	S	H	H	S	-
3	O	H	H	W	+
4	R	M	H	W	+
5	R	C	N	W	+
6	R	C	N	S	-
7	O	C	N	S	+
8	S	M	H	W	-
9	S	C	N	W	+
10	R	M	N	W	+
11	S	M	N	S	+
12	O	M	H	S	+
13	O	H	N	W	+
14	R	M	H	S	-

Outlook: S(unny), O(vercast), R(ainy)

Te

Hu

N(ormal),
L(ow)

Wind: S(trong), W(eak)

We need to learn

1. The prior $P(\text{Play?})$
2. The likelihoods $P(\text{x} \mid \text{Play?})$

How hard is it to learn probabilistic models?

	O	T	H	W	Play?
1	S	H	H	W	-
2	S	H	H	S	-
3	O	H	H	W	+
4	R	M	H	W	+
5	R	C	N	W	+
6	R	C	N	S	-
7	O	C	N	S	+
8	S	M	H	W	-
9	S	C	N	W	+
10	R	M	N	W	+
11	S	M	N	S	+
12	O	M	H	S	+
13	O	H	N	W	+
14	R	M	H	S	-

Prior P(play?)

- A single number (Why only one?)

How hard is it to learn probabilistic models?

	O	T	H	W	Play?
1	S	H	H	W	-
2	S	H	H	S	-
3	O	H	H	W	+
4	R	M	H	W	+
5	R	C	N	W	+
6	R	C	N	S	-
7	O	C	N	S	+
8	S	M	H	W	-
9	S	C	N	W	+
10	R	M	N	W	+
11	S	M	N	S	+
12	O	M	H	S	+
13	O	H	N	W	+
14	R	M	H	S	-

Prior P(play?)

- A single number (Why only one?)

Likelihood P(**X** | Play?)

- There are 4 features
- For each value of Play? (+/-), we need a value for each possible assignment: P(O, T, H, W | Play?)

How hard is it to learn probabilistic models?

	O	T	H	W	Play?
1	S	H	H	W	-
2	S	H	H	S	-
3	O	H	H	W	+
4	R	M	H	W	+
5	R	C	N	W	+
6	R	C	N	S	-
7	O	C	N	S	+
8	S	M	H	W	-
9	S	C	N	W	+
10	R	M	N	W	+
11	S	M	N	S	+
12	O	M	H	S	+
13	O	H	N	W	+
14	R	M	H	S	-
	3	3	3	2	

Values for this feature

Prior P(play?)

- A single number (Why only one?)

Likelihood P(**X** | Play?)

- There are 4 features
- For each value of Play? (+/-), we need a value for each possible assignment: P(O, T, H, W | Play?)

How hard is it to learn probabilistic models?

	O	T	H	W	Play?
1	S	H	H	W	-
2	S	H	H	S	-
3	O	H	H	W	+
4	R	M	H	W	+
5	R	C	N	W	+
6	R	C	N	S	-
7	O	C	N	S	+
8	S	M	H	W	-
9	S	C	N	W	+
10	R	M	N	W	+
11	S	M	N	S	+
12	O	M	H	S	+
13	O	H	N	W	+
14	R	M	H	S	-
	3	3	3	2	

Values for this feature

Prior P(play?)

- A single number (Why only one?)

Likelihood P(**X** | Play?)

- There are 4 features
- For each value of Play? (+/-), we need a value for each possible assignment: P(O, T, H, W | Play?)
- $(3 \cdot 3 \cdot 3 \cdot 2 \ - 1)$ parameters in each case

One for each assignment

How hard is it to learn probabilistic models?

	O	T	H	W	Play?
1	S	H	H	W	-
2	S	H	H	S	-
3	O	H	H	W	+
4	R	M	H	W	+
5	R	C	N	W	+
6	R	C	N	S	-
7	O	C	N	S	+
8	S	M	H	W	-
9	S	C	N	W	+
10	R	M	N	W	+
11	S	M	N	S	+
12	O	M	H	S	+
13	O	H	N	W	+
14	R	M	H	S	-

In general

Prior P(Y)

- If there are k labels, then k – 1 parameters (why not k?)

How hard is it to learn probabilistic models?

	O	T	H	W	Play?
1	S	H	H	W	-
2	S	H	H	S	-
3	O	H	H	W	+
4	R	M	H	W	+
5	R	C	N	W	+
6	R	C	N	S	-
7	O	C	N	S	+
8	S	M	H	W	-
9	S	C	N	W	+
10	R	M	N	W	+
11	S	M	N	S	+
12	O	M	H	S	+
13	O	H	N	W	+
14	R	M	H	S	-

In general

Prior P(Y)

- If there are k labels, then k – 1 parameters (why not k?)

Likelihood P(**X** | Y)

- If there are d Boolean features:
  - We need a value for each possible $P(x_1, x_2, \cdots, x_d \mid y)$ for each y
  - $k(2^d - 1)$ parameters

How hard is it to learn probabilistic models?

	O	T	H	W	Play?
1	S	H	H	W	-
2	S	H	H	S	-
3	O	H	H	W	+
4	R	M	H	W	+
5	R	C	N	W	+
6	R	C	N	S	-
7	O	C	N	S	+
8	S	M	H	W	-
9	S	C	N	W	+
10	R	M	N	W	+
11	S	M	N	S	+
12	O	M	H	S	+
13	O	H	N	W	+
14	R	M	H	S	-

In general

Prior P(Y)

- If there are k labels, then k – 1 parameters (why not k?)

Likelihood P(**X** | Y)

- If there are d Boolean features:
  - We need a value for each possible $P(x_1, x_2, \cdots, x_d \mid y)$ for each y
  - $k(2^d – 1)$ parameters

Need a lot of data to estimate these many numbers!

How hard is it to learn probabilistic models?

Prior P(Y)

- If there are k labels, then k – 1 parameters (why not k?)

Likelihood P(**X** | Y)

- If there are d Boolean features:
  - We need a value for each possible $P(x_1, x_2, \cdots, x_d \mid y)$ for each y
  - $k(2^d - 1)$ parameters

Need a lot of data to estimate these many numbers!

High model complexity

If there is very limited data, high variance in the parameters

How hard is it to learn probabilistic models?

Prior P(Y)

- If there are k labels, then k – 1 parameters (why not k?)

Likelihood P(**X** | Y)

- If there are d Boolean features:
  - We need a value for each possible $P(x_1, x_2, \cdots, x_d \mid y)$ for each y
  - $k(2^d - 1)$ parameters

Need a lot of data to estimate these many numbers!

High model complexity

If there is very limited data, high variance in the parameters

How can we deal with this?

How hard is it to learn probabilistic models?

Prior P(Y)

- If there are k labels, then $k - 1$ parameters (why not k?)

Likelihood P(**X** | Y)

- If there are d Boolean features:
  - We need a value for each possible $P(x_1, x_2, \cdots, x_d \mid y)$ for each y
  - $k(2^d - 1)$ parameters

Need a lot of data to estimate these many numbers!

High model complexity

If there is very limited data, high variance in the parameters

How can we deal with this?

Answer: Make independence assumptions

Recall: Conditional independence

Suppose X, Y and Z are random variables

X is *conditionally independent* of Y given Z if the probability distribution of X is independent of the value of Y when Z is observed

$$P(X|Y,Z) = P(X|Z)$$

Or equivalently

$$P(X,Y|Z) = P(X|Z)P(Y|Z)$$

Modeling the features

$P(x_1, x_2, \cdots, x_d | y)$ required k(2^d – 1) parameters

What if *all the features were conditionally independent given the label*? *The Naïve Bayes Assumption*

Modeling the features

$P(x_1, x_2, \cdots, x_d | y)$ required k(2^{d} – 1) parameters

What if *all the features were conditionally independent given the label*? *The Naïve Bayes Assumption*

That is,

$$P(x_1, x_2, \cdots, x_d | y) = P(x_1 | y) P(x_2 | y) \cdots P(x_d | y)$$

Requires only d numbers for each label. kd parameters overall. Not bad!

The Naïve Bayes Classifier

Assumption: Features are conditionally independent given the label Y

To predict, we need two sets of probabilities

- Prior $P(y)$
- For each x_j, we have the likelihood $P(x_j \mid y)$

The Naïve Bayes Classifier

Assumption: Features are conditionally independent given the label Y

To predict, we need two sets of probabilities

- Prior P(y)
- For each x_j, we have the likelihood $P(x_j \mid y)$

Decision rule

$$h_{NB}(\boldsymbol{x}) = \operatorname*{argmax}_{y} P(y)P(x_1, x_2, \cdots, x_d | y)$$

The Naïve Bayes Classifier

Assumption: Features are conditionally independent given the label Y

To predict, we need two sets of probabilities

- Prior P(y)
- For each x_j, we have the likelihood $P(x_j \mid y)$

Decision rule

$$\begin{aligned} h_{NB}(\boldsymbol{x}) &= \underset{y}{\operatorname{argmax}}\, P(y) P(x_1, x_2, \cdots, x_d | y) \\ &= \underset{y}{\operatorname{argmax}}\, P(y) \prod_j P(x_j | y) \end{aligned}$$

Decision boundaries of naïve Bayes

What is the decision boundary of the naïve Bayes classifier?

Consider the two class case. We predict the label to be + if

$$P(y = +) \prod_j P(x_j | y = +) > P(y = -) \prod_j P(x_j | y = -)$$

Decision boundaries of naïve Bayes

What is the decision boundary of the naïve Bayes classifier?

Consider the two class case. We predict the label to be + if

$$P(y=+)\prod_j P(x_j|y=+) > P(y=-)\prod_j P(x_j|y=-)$$

$$\frac{P(y=+)\prod_j P(x_j|y=+)}{P(y=-)\prod_j P(x_j|y=-)} > 1$$

Decision boundaries of naïve Bayes

What is the decision boundary of the naïve Bayes classifier?

Taking log and simplifying, we get

$$\log \frac{P(y = -|\boldsymbol{x})}{P(y = +|\boldsymbol{x})} = \boldsymbol{w}^T \boldsymbol{x} + b$$

This is a linear function of the feature space!

Easy to prove. See note on course website

Today’s lecture

- The naïve Bayes Classifier
- Learning the naïve Bayes Classifier
- Practical Concerns

Learning the naïve Bayes Classifier

- What is the hypothesis function *h* defined by?

Learning the naïve Bayes Classifier

- What is the hypothesis function *h* defined by?
  - A collection of probabilities
    - Prior for each label: $P(y)$
    - Likelihoods for feature x_j given a label: $P(x_j | y)$

Learning the naïve Bayes Classifier

- What is the hypothesis function *h* defined by?
  - A collection of probabilities
    - Prior for each label: $P(y)$
    - Likelihoods for feature x_j given a label: $P(x_j | y)$

Suppose we have a data set $D = \{(\boldsymbol{x}_i, y_i)\}$ with m examples

Learning the naïve Bayes Classifier

- What is the hypothesis function *h* defined by?
  - A collection of probabilities
    - Prior for each label: $P(y)$
    - Likelihoods for feature x_j given a label: $P(x_j|\ y)$

Suppose we have a data set $D = \{(\boldsymbol{x}_i, y_i)\}$ with m examples

A note on convention for this section:

- Examples in the dataset are indexed by the subscript i (e.g. $\boldsymbol{x}_i$)
- Features within an example are indexed by the subscript j
  - The j^{th} feature of the i^{th} example will be x_{ij}

Learning the naïve Bayes Classifier

- What is the hypothesis function h defined by?
  - A collection of probabilities
    - Prior for each label: $P(y)$
    - Likelihoods for feature x_j given a label: $P(x_j|\, y)$

If we have a data set $D = \{(\boldsymbol{x}_i, y_i)\}$ with m examples

And we want to learn the classifier in a probabilistic way

– What is a probabilistic criterion to select the hypothesis?

Learning the naïve Bayes Classifier

Maximum likelihood estimation

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

Here h is defined by all the probabilities used to construct the naïve Bayes decision

Maximum likelihood estimation

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

Given a dataset $D = \{(\boldsymbol{x}_i, y_i)\}$ with m examples

$$h_{ML} = \arg\max_{h} \prod_{i=1}^{m} P((\mathbf{x}_i, y_i)|h)$$

Each example in the dataset is independent and identically distributed

So we can represent $P(D|\ h)$ as this product

Maximum likelihood estimation

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

Given a dataset $D = \{(\boldsymbol{x}_i, y_i)\}$ with m examples

$$h_{ML} = \arg\max_{h} \prod_{i=1}^{m} \boxed{P((\mathbf{x}_i, y_i)|h)}$$

Each example in the dataset is independent and identically distributed

So we can represent $P(D|\ h)$ as this product

Asks "What probability would this particular h assign to the pair $(\mathbf{x}_i, y_i)$?"

Maximum likelihood estimation

Given a dataset D = {(x_i, y_i)} with m examples

$$\begin{aligned} h_{ML} &= \arg\max_h \prod_{i=1}^{m} P((\mathbf{x}_i, y_i)|h) \\ &= \arg\max_h \prod_{i=1}^{m} P(\mathbf{x}_i|y_i, h)P(y_i|h) \end{aligned}$$

Maximum likelihood estimation

Given a dataset D = {(x_i, y_i)} with m examples

$$
\begin{aligned}
h_{ML} &= \arg\max_h \prod_{i=1}^{m} P((\mathbf{x}_i, y_i)|h) \\
&= \arg\max_h \prod_{i=1}^{m} P(\mathbf{x}_i|y_i, h) P(y_i|h) \\
&= \arg\max_h \prod_{i=1}^{m} P(y_i|h) \prod_j P(x_{i,j}|y_i, h)
\end{aligned}
$$

$\boldsymbol{x}_{ij}$ is the j^th feature of $\mathbf{x}_i$

The Naïve Bayes assumption

Maximum likelihood estimation

Given a dataset D = {(x_i, y_i)} with m examples

$$\begin{aligned} h_{ML} &= \arg\max_h \prod_{i=1}^{m} P((\mathbf{x}_i, y_i)|h) \\ &= \arg\max_h \prod_{i=1}^{m} P(\mathbf{x}_i|y_i, h)P(y_i|h) \\ &= \arg\max_h \prod_{i=1}^{m} P(y_i|h) \prod_j P(x_{i,j}|y_i, h) \end{aligned}$$

How do we proceed?

Maximum likelihood estimation

Given a dataset D = {(x_i, y_i)} with m examples

$$
\begin{aligned}
h_{ML} &= \arg\max_h \prod_{i=1}^{m} P((\mathbf{x}_i, y_i)|h) \\
&= \arg\max_h \prod_{i=1}^{m} P(\mathbf{x}_i|y_i, h)P(y_i|h) \\
&= \arg\max_h \prod_{i=1}^{m} P(y_i|h) \prod_j P(x_{i,j}|y_i, h) \\
&= \arg\max_h \sum_{i=1}^{m} \log P(y_i|h) + \sum_i \sum_j \log P(x_{i,j}|y_i, h)
\end{aligned}
$$

Learning the naïve Bayes Classifier

Maximum likelihood estimation

$$h_{ML} = \arg\max_h \sum_{i=1}^{m} \log P(y_i|h) + \sum_i \sum_j \log P(x_{i,j}|y_i, h)$$

What next?

Learning the naïve Bayes Classifier

Maximum likelihood estimation

$$h_{ML} = \arg\max_h \sum_{i=1}^{m} \log P(y_i|h) + \sum_i \sum_j \log P(x_{i,j}|y_i, h)$$

What next?

We need to make a modeling assumption about the functional form of these probability distributions

Learning the naïve Bayes Classifier

Maximum likelihood estimation

$$h_{ML} = \arg\max_h \sum_{i=1}^{m} \log P(y_i|h) + \sum_i \sum_j \log P(x_{i,j}|y_i, h)$$

For simplicity, suppose there are two labels 1 and 0 and all features are binary

- **Prior**: P(y = 1) = p and P (y = 0) = 1 – p

That is, the prior probability is from the Bernoulli distribution.

Learning the naïve Bayes Classifier

Maximum likelihood estimation

$$h_{ML} = \arg\max_h \sum_{i=1}^{m} \log P(y_i|h) + \sum_i \sum_j \log P(x_{i,j}|y_i, h)$$

For simplicity, suppose there are two labels 1 and 0 and all features are binary

- **Prior**: P(y = 1) = p and P (y = 0) = 1 – p
- **Likelihood** for each feature given a label
  - P(x_j = 1 | y = 1) = a_j and P(x_j = 0 | y = 1) = 1 – a_j

Learning the naïve Bayes Classifier

Maximum likelihood estimation

$$h_{ML} = \arg\max_h \sum_{i=1}^{m} \log P(y_i|h) + \sum_i \sum_j \log P(x_{i,j}|y_i, h)$$

For simplicity, suppose there are two labels 1 and 0 and all features are binary

- **Prior**: P(y = 1) = p and P (y = 0) = 1 – p
- **Likelihood** for each feature given a label
  - $P(x_j = 1 \mid y = 1) = a_j$ and $P(x_j = 0 \mid y = 1) = 1 - a_j$
  - $P(x_j = 1 \mid y = 0) = b_j$ and $P(x_j = 0 \mid y = 0) = 1 - b_j$

That is, the likelihood of each feature is also is from the Bernoulli distribution.

Learning the naïve Bayes Classifier

Maximum likelihood estimation

$$h_{ML} = \arg\max_h \sum_{i=1}^{m} \log P(y_i|h) + \sum_i \sum_j \log P(x_{i,j}|y_i, h)$$

For simplicity, suppose there are two labels 1 and 0 and all features are binary

- **Prior**: P(y = 1) = p and P (y = 0) = 1 – p
- **Likelihood** for each feature given a label
  - $P(x_j = 1 \mid y = 1) = a_j$ and $P(x_j = 0 \mid y = 1) = 1 - a_j$
  - $P(x_j = 1 \mid y = 0) = b_j$ and $P(x_j = 0 \mid y = 0) = 1 - b_j$

h consists of p, all the a's and b's

Learning the naïve Bayes Classifier

Maximum likelihood estimation

$$h_{ML} = \arg\max_h \sum_{i=1}^{m} \log P(y_i|h) + \sum_i \sum_j \log P(x_{i,j}|y_i, h)$$

- Prior: P(y = 1) = p and P (y = 0) = 1 – p

Learning the naïve Bayes Classifier

Maximum likelihood estimation

$$h_{ML} = \arg\max_h \sum_{i=1}^{m} \log P(y_i|h) + \sum_i \sum_j \log P(x_{i,j}|y_i, h)$$

- Prior: P(y = 1) = p and P (y = 0) = 1 – p

$$P(y_i|h) = p^{[y_i=1]}(1-p)^{[y_i=0]}$$

[z] is called the indicator function or the Iverson bracket

Its value is 1 if the argument z is true and zero otherwise

Learning the naïve Bayes Classifier

Maximum likelihood estimation

$$h_{ML} = \arg\max_h \sum_{i=1}^{m} \log P(y_i|h) + \sum_i \sum_j \log P(x_{i,j}|y_i, h)$$

Likelihood for each feature given a label

- $P(x_j = 1 \mid y = 1) = a_j$ and $P(x_j = 0 \mid y = 1) = 1 - a_j$
- $P(x_j = 1 \mid y = 0) = b_j$ and $P(x_j = 0 \mid y = 0) = 1 - b_j$

$$P(x_{ij}|y_i, h) = a_j^{[y_i=1, x_{ij}=1]} \times (1 - a_j)^{[y_i=1, x_{ij}=0]} \times b_j^{[y_i=0, x_{ij}=1]} \times (1 - b_j)^{[y_i=0, x_{ij}=0]}$$

Learning the naïve Bayes Classifier

Substituting and deriving the argmax, we get

$$p = \frac{\text{Count}(y_i = 1)}{\text{Count}(y_i = 1) + \text{Count}(y_i = 0)}$$

← P(y = 1) = p

Learning the naïve Bayes Classifier

Substituting and deriving the argmax, we get

$$p = \frac{\text{Count}\,(y_i = 1)}{\text{Count}\,(y_i = 1) + \text{Count}\,(y_i = 0)}$$

P(y = 1) = p

$$a_j = \frac{\text{Count}\,(y_i = 1, x_{ij} = 1)}{\text{Count}\,(y_i = 1)}$$

P(x_j = 1 | y = 1) = a_j

Learning the naïve Bayes Classifier

Substituting and deriving the argmax, we get

$$p = \frac{\text{Count}\left(y_i = 1\right)}{\text{Count}\left(y_i = 1\right) + \text{Count}\left(y_i = 0\right)}$$

← P(y = 1) = p

$$a_j = \frac{\text{Count}\left(y_i = 1, x_{ij} = 1\right)}{\text{Count}\left(y_i = 1\right)}$$

← P(x_j = 1 | y = 1) = a_j

$$b_j = \frac{\text{Count}\left(y_i = 0, x_{ij} = 1\right)}{\text{Count}\left(y_i = 0\right)}$$

← P(x_j = 1 | y = 0) = b_j

Let's learn a naïve Bayes classifier

	O	T	H	W	Play?
1	S	H	H	W	-
2	S	H	H	S	-
3	O	H	H	W	+
4	R	M	H	W	+
5	R	C	N	W	+
6	R	C	N	S	-
7	O	C	N	S	+
8	S	M	H	W	-
9	S	C	N	W	+
10	R	M	N	W	+
11	S	M	N	S	+
12	O	M	H	S	+
13	O	H	N	W	+
14	R	M	H	S	-

With the assumption that all our probabilities are from the Bernoulli distribution

Let's learn a naïve Bayes classifier

	O	T	H	W	Play?
1	S	H	H	W	-
2	S	H	H	S	-
3	O	H	H	W	+
4	R	M	H	W	+
5	R	C	N	W	+
6	R	C	N	S	-
7	O	C	N	S	+
8	S	M	H	W	-
9	S	C	N	W	+
10	R	M	N	W	+
11	S	M	N	S	+
12	O	M	H	S	+
13	O	H	N	W	+
14	R	M	H	S	-

$$P(Play = +) = \frac{9}{14} \qquad P(Play = -) = \frac{5}{14}$$

Let’s learn a naïve Bayes classifier

	O	T	H	W	Play?
1	S	H	H	W	-
2	S	H	H	S	-
3	O	H	H	W	+
4	R	M	H	W	+
5	R	C	N	W	+
6	R	C	N	S	-
7	O	C	N	S	+
8	S	M	H	W	-
9	S	C	N	W	+
10	R	M	N	W	+
11	S	M	N	S	+
12	O	M	H	S	+
13	O	H	N	W	+
14	R	M	H	S	-

$$P(Play\ =\ +) = \frac{9}{14} \qquad P(Play\ =\ -) = \frac{5}{14}$$

$$P(\boldsymbol{O}\ =\ S \mid Play\ =\ +) = \frac{2}{9}$$

Let's learn a naïve Bayes classifier

	O	T	H	W	Play?
1	S	H	H	W	-
2	S	H	H	S	-
3	O	H	H	W	+
4	R	M	H	W	+
5	R	C	N	W	+
6	R	C	N	S	-
7	O	C	N	S	+
8	S	M	H	W	-
9	S	C	N	W	+
10	R	M	N	W	+
11	S	M	N	S	+
12	O	M	H	S	+
13	O	H	N	W	+
14	R	M	H	S	-

$$P(Play = +) = \frac{9}{14} \qquad P(Play = -) = \frac{5}{14}$$

$$P(\boldsymbol{O} = S \mid Play = +) = \frac{2}{9}$$

$$P(\boldsymbol{O} = R \mid Play = +) = \frac{3}{9}$$

Let's learn a naïve Bayes classifier

	O	T	H	W	Play?
1	S	H	H	W	-
2	S	H	H	S	-
3	O	H	H	W	+
4	R	M	H	W	+
5	R	C	N	W	+
6	R	C	N	S	-
7	O	C	N	S	+
8	S	M	H	W	-
9	S	C	N	W	+
10	R	M	N	W	+
11	S	M	N	S	+
12	O	M	H	S	+
13	O	H	N	W	+
14	R	M	H	S	-

$$P(Play = +) = \frac{9}{14} \qquad P(Play = -) = \frac{5}{14}$$

$$P(\boldsymbol{O} = S \mid Play = +) = \frac{2}{9}$$

$$P(\boldsymbol{O} = R \mid Play = +) = \frac{3}{9}$$

$$P(\boldsymbol{O} = O \mid Play = +) = \frac{4}{9}$$

And so on, for other attributes and also for Play = -

Naïve Bayes: Learning and Prediction

- Learning
  - Count how often features occur with each label. Normalize to get likelihoods
  - Priors from fraction of examples with each label
  - Generalizes to multiclass
- Prediction
  - Use learned probabilities to find highest scoring label

Today’s lecture

- The naïve Bayes Classifier
- Learning the naïve Bayes Classifier
- Practical concerns + an example

Important caveats with Naïve Bayes

1. Features need not be conditionally independent given the label
   - Just because we assume that they are doesn't mean that that's how they behave in nature
   - We made a modeling assumption because it makes computation and learning easier

2. Not enough training data to get good estimates of the probabilities from counts

Important caveats with Naïve Bayes

1. Features are not conditionally independent given the label

All bets are off if the naïve Bayes assumption is not satisfied

$$P(\mathbf{x}|y) \neq \prod P(x_j|y)$$

And yet, very often used in practice because of simplicity

Works reasonably well even when the assumption is violated

Important caveats with Naïve Bayes

2. Not enough training data to get good estimates of the probabilities from counts

The basic operation for learning likelihoods is counting how often a feature occurs with a label.

What if we never see a particular feature with a particular label?

Eg: Suppose we never observe Temperature = cold with PlayTennis= Yes

Should we treat those counts as zero?

Important caveats with Naïve Bayes

2. Not enough training data to get good estimates of the probabilities from counts

The basic operation for learning likelihoods is counting how often a feature occurs with a label.

What if we never see a particular feature with a particular label?

Eg: Suppose we never observe Temperature = cold with PlayTennis= Yes

Should we treat those counts as zero? But that will make the probabilities zero

Important caveats with Naïve Bayes

2. Not enough training data to get good estimates of the probabilities from counts

The basic operation for learning likelihoods is counting how often a feature occurs with a label.

What if we never see a particular feature with a particular label?

Eg: Suppose we never observe Temperature = cold with PlayTennis= Yes

Should we treat those counts as zero? But that will make the probabilities zero

Answer: Smoothing

- Add fake counts (very small numbers so that the counts are not zero)
- The Bayesian interpretation of smoothing: Priors on the hypothesis (MAP learning)

Example: Classifying text

- Instance space: Text documents
- Labels: Spam or NotSpam

- Goal: To learn a function that can predict whether a new document is Spam or NotSpam

How would you build a Naïve Bayes classifier?

Let us brainstorm

How to represent documents?
How to estimate probabilities?
How to classify?

Example: Classifying text

1. Represent documents by a vector of words
   A sparse vector consisting of one feature per word

Example: Classifying text

1. Represent documents by a vector of words

    A sparse vector consisting of one feature per word

2. Learning from N labeled documents

Example: Classifying text

1. Represent documents by a vector of words

   A sparse vector consisting of one feature per word

2. Learning from N labeled documents
   1. Priors $P(\text{Spam}) = \frac{\text{Count}\ (\text{Spam})}{N}; P(\text{NotSpam}) = 1 - P(\text{Spam})$

Example: Classifying text

1. Represent documents by a vector of words

   A sparse vector consisting of one feature per word

2. Learning from N labeled documents
   1. Priors $P(\text{Spam}) = \frac{\text{Count}\,(\text{Spam})}{N}; P(\text{NotSpam}) = 1 - P(\text{Spam})$
   2. For each word w in vocabulary :

$$P(\text{w}|\text{Spam}) = \frac{\text{Count}\,(\text{w}, \text{Spam}) + 1}{\text{Count}\,(\text{Spam}) + |\text{Vocabulary}|}$$

$$P(\text{w}|\text{NotSpam}) = \frac{\text{Count}\,(\text{w}, \text{NotSpam}) + 1}{\text{Count}\,(\text{NotSpam}) + |\text{Vocabulary}|}$$

Example: Classifying text

1. Represent documents by a vector of words

   A sparse vector consisting of one feature per word

2. Learning from N labeled documents

   1. Priors $P(\text{Spam}) = \frac{\text{Count}\,(\text{Spam})}{N}; P(\text{NotSpam}) = 1 - P(\text{Spam})$

   2. For each word w in vocabulary :

$$P(\text{w}|\text{Spam}) = \frac{\text{Count}\,(\text{w}, \text{Spam}) + 1}{\text{Count}\,(\text{Spam}) + |\text{Vocabulary}|}$$

$$P(\text{w}|\text{NotSpam}) = \frac{\text{Count}\,(\text{w}, \text{NotSpam}) + 1}{\text{Count}\,(\text{NotSpam}) + |\text{Vocabulary}|}$$

How often does a word occur with a label?

Example: Classifying text

1. Represent documents by a vector of words

   A sparse vector consisting of one feature per word

2. Learning from N labeled documents

   1. Priors $P(\text{Spam}) = \frac{\text{Count}\,(\text{Spam})}{N}; P(\text{NotSpam}) = 1 - P(\text{Spam})$

   2. For each word w in vocabulary :

Smoothing

$$P(\text{w}|\text{Spam}) = \frac{\text{Count}\,(\text{w}, \text{Spam}) + 1}{\text{Count}\,(\text{Spam}) + |\text{Vocabulary}|}$$

$$P(\text{w}|\text{NotSpam}) = \frac{\text{Count}\,(\text{w}, \text{NotSpam}) + 1}{\text{Count}\,(\text{NotSpam}) + |\text{Vocabulary}|}$$

Continuous features

- So far, we have been looking at discrete features
  - $P(x_j \mid y)$ is a Bernoulli trial (i.e. a coin toss)

- We could model $P(x_j \mid y)$ with other distributions too
  - This is a separate assumption from the independence assumption that naive Bayes makes
  - Eg: For real valued features, $(X_j \mid Y)$ could be drawn from a normal distribution

- **Exercise**: Derive the maximum likelihood estimate when the features are assumed to be drawn from the normal distribution

Summary: Naïve Bayes

- Independence assumption
  - All features are independent of each other given the label
- Maximum likelihood learning: Learning is simple
  - Generalizes to real valued features
- Prediction via MAP estimation
  - Generalizes to beyond binary classification
- Important caveats to remember
  - Smoothing
  - Independence assumption may not be valid
- Decision boundary is linear for binary classification