Naïve Bayes Classifier

Lecture 12

Unbiased Learning of Bayes Classifiers is Impractical

- Learn Bayes classifier by estimating $P(X|Y)$ and $P(Y)$.
- Assume Y is boolean and X is a vector of *n* boolean attributes. In this case, we need to estimate a set of parameters $\theta_{ij} \equiv P\left(X = x_i \mid Y = y_j\right)$

  i takes on 2^n possible values; j takes on 2 possible values.
- How many parameters?
  - For any particular value y_j, and the 2^n possible values of x_i, we need compute 2^n-1 independent parameters.
  - Given the two possible values for *Y*, we must estimate a total of $2(2^n$-1) such parameters.

Complex model →High variance with limited data!!!

Conditional Independence

- X is **conditionally independent** of Y given Z, if the probability distribution governing X is independent of the value of Y, given the value of Z

$$(\forall i, j, k)\ P\left(X = x_i \mid Y = y_j, Z = z_k\right) = P\left(X = x_i \mid Z = z_k\right)$$

- Example:

$$P(\text{Thunder} \mid \text{Rain}, \text{Lighting}) = P(\text{Thunder} \mid \text{Lighting})$$

Note that in general Thunder is not independent of Rain, but it is given Lighting.

- Equivalent to:

$$P(X, Y \mid Z) = P(X \mid Z) P(Y \mid Z)$$

Derivation of Naive Bayes Algorithm

- Naive Bayes algorithm assumes that the attributes $X_1,\ldots,X_n$ are all conditionally independent of one another, given Y. This dramatically simplifies
  - the representation of $P(X|Y)$
  - estimating $P(X|Y)$ from the training data.
- Consider $X=(X_1,X_2)$

$$P(X \mid Y) = P(X_1, X_2 \mid Y) = P(X_1 \mid Y)P(X_2 \mid Y)$$

- For X containing n attributes

$$P(X \mid Y) = \prod_{i=1}^{n} P(X_i \mid Y)$$

Given the boolean X and Y, now we need only 2*n* parameters to define $P(X|Y)$, which is dramatic reduction compared to the $2(2^n-1)$ parameters if we make no conditional independence assumption.

The Naïve Bayes Classifier

- Given:
  - Prior $P(Y)$
  - *n* conditionally independent features X, given the class Y
  - For each X_i, we have likelihood $P(X_i|Y)$
- The probability that Y will take on its kth possible value, is

$$P(Y = y_k \mid X) = \frac{P(Y = y_k)\prod_i P(X_i \mid Y = y_k)}{\sum_j P(Y = y_k)\prod_i P(X_i \mid Y = y_k)}$$

- The Decision rule:

$$y^* = \arg\max_{y_k} P(Y = y_k)\prod_i P(X_i \mid Y = y_k)$$

If assumption holds, NB is optimal classifier!

Naïve Bayes for the discrete inputs

- Given, n attributes X_i each taking on J possible discrete values and Y a discrete variable taking on K possible values.
- MLE for Likelihood $P(X_i = x_{ij} \mid Y = y_k)$ given a set of training examples *D:*

$$\hat{P}(X_i = x_{ij} \mid Y = y_k) = \frac{\#D\{X_i = x_{ij} \wedge Y = y_k\}}{\#D\{Y = y_k\}}$$

where the *#D{x}* operator returns the number of elements in the set *D* that satisfy property *x*.

- MLE for the prior

$$\hat{P}(Y = y_k) = \frac{\#D\{Y = y_k\}}{|D|}$$

number of elements in the training set *D*

NB Example

- Given, training data

Day	*Outlook*	*Temperature*	*Humidity*	*Wind*	*PlayTennis*
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

- Classify the following novel instance :

(Outlook=sunny, Temp=cool,Humidity=high,Wind=strong)

NB Example

$$y_{NB} = \underset{y_j = \{yes,no\}}{\arg\max} P(y_j) P(Outlook = sunny \mid y_j) P(Temp = cool \mid y_j)$$

$$P(Humidity = high \mid y_j) P(Wind = strong \mid y_j)$$

Priors:

$$P(PlayTennis = yes) = 9/14 = 0.64$$

$$P(PlayTennis = no) = 5/14 = 0.36$$

$\text{Conditional Probabilities, e.g. Wind = strong:}$

$$P(Wind = strong \mid PlayTennis = yes) = 3/9 = 0.33$$

$$P(Wind = strong \mid PlayTennis = no) = 3/5 = 0.6$$

...

$$P(yes)P(sunny \mid yes)P(cool \mid yes)P(high \mid yes)P(strong \mid yes) = 0.0053$$

$$\boxed{P(no)P(sunny \mid no)P(cool \mid no)P(high \mid no)P(strong \mid no) = 0.60}$$

Subtleties of NB classifier 1 –Violating the NB assumption

- Usually, features are not conditionally independent.
- Nonetheless, NB often performs well, even when assumption is violated
  - [Domingos& Pazzani’96] discuss some conditions for good performance

Subtleties of NB classifier 2 – Insufficient training data

- What if you never see a training instance where X_1=a when Y=b?
  - e.g., Y={SpamEmail}, X_1={'Market'}
  - $P(X_1=a \mid Y=b) = 0$
- Thus, no matter what the values $X_2,\ldots,X_n$ take:

$$P(Y=b \mid X_1=a, X_2, \ldots, X_n) = 0$$

- Solution?

Subtleties of NB classifier 2 – Insufficient training data

- To avoid this, use a "smoothed" estimate
  - effectively adds in a number of additional "hallucinated" examples
  - assumes these hallucinated examples are spread evenly over the possible values of X_i.
- This smoothed estimate is given by

$$\hat{P}(X_i = x_{ij} \mid Y = y_k) = \frac{\# D\{X_i = x_{ij} \wedge Y = y_k\} + l}{\# D\{Y = y_k\} + lJ}$$

$$\hat{P}(Y = y_k) = \frac{\# D\{Y = y_k\} + l}{|D| + lK}$$

The number of hallucinated examples

l determines the strength of the smoothing

If *l=1* called Laplace smoothing

Cont…

- Denote

$$\hat{P}(X_i = x_{ij} \mid Y = y_k) = \frac{\overbrace{\#D\{X_i = x_{ij} \wedge Y = y_k\}}^{n_c} + l}{\underbrace{\#D\{Y = y_k\}}_{n} + \underbrace{lJ}_{m}}$$

- Then

$$\hat{P}(X_i = x_{ij} \mid Y = y_k) = \frac{n_c + m\dfrac{1}{J}}{n+m}$$

- We can view it as a Bayesian approach to estimating the probability: $\hat{P}(X_i = x_{ij} \mid Y = y_k) = \dfrac{n_c + mp}{n+m}$

  with uniform prior $p = \dfrac{1}{J}$

- The observed fraction and prior are combined with the weight m.

Naive Bayes for Continuous Inputs

- When the X_i are continuous we must choose some other way to represent the distributions $P(X_i|Y)$.
- One common approach is to assume that for each possible discrete value y_k of Y, the distribution of each continuous X_i is Gaussian.
- In order to train such a Naïve Bayes classifier we must estimate the mean and standard deviation of each of these Gaussians

Naive Bayes for Continuous Inputs

- MLE for means

$$\hat{\mu}_{ik} = \frac{1}{\sum_j \delta(Y^j = y_k)} \sum_j X_i^j \delta(Y^j = y_k)$$

  - where j refers to the jth training example, and where $\delta(Y=y_k)$ is 1 if $Y = y_k$ and 0 otherwise.
  - Note the role of δ is to select only those training examples for which $Y = y_k$.

- MLE for standard deviation

$$\hat{\sigma}_{ik}^2 = \frac{1}{\sum_j \delta(Y^j = y_k)} \sum_j \left(X_i^j - \hat{\mu}_{ik}\right)^2 \delta(Y^j = y_k)$$

Learning Classify Text

- Applications:
  - Learn which news article are of interest
  - Learn to classify web pages by topic.
- Naïve Bayes is among most effective algorithms

- Target concept *Interesting?: Document*->{+,-}

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|-)

Text Classification-Example:

Text

Text Classification, or the task of automatically assigning semantic categories to natural language text, has become one of the key methods for organizing online information. Since hand-coding classification rules is costly or even impractical, most modern approaches employ machine learning techniques to automatically learn text classifiers from examples.

Text Representation

(a_1='text',a_2='classification',….
a_{48}='examples')

The text contains 48 words

The representation contains 48 attributes

Note: Text size may vary, but it will not cause a problem

NB conditional independence Assumption

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

probability that word in position i is w_k, given y_j

Indicates the kth word in English vocabulary

The NB assumption is that the word probabilities for one text position are independent of the words in other positions, given the document classification y_j

Clearly not true: The probability of word "learning" may be greater if the preceding word is "machine"

Necessary, without it the number of probability terms is prohibitive

Performs remarkably well despite the incorrectness of the assumption

Text Classification-Example:

Text

Text Classification, or the task of automatically assigning semantic categories to natural language text, has become one of the key methods for organizing online information. Since hand-coding classification rules is costly or even impractical, most modern approaches employ machine learning techniques to automatically learn text classifiers from examples.

Text Representation

(a_1='text',a_2='classification',….
a_{48}='examples')

The text contains 48 words

The representation contains 48 attributes

Classification:

$$y^* = \arg\max_{y_j \in \{+,-\}} P(y_j) P(a_1 = 'text' \mid y_j) \ldots P(a_{48} = 'examples' \mid y_j)$$

$$= \arg\max_{y_j \in \{+,-\}} P(y_j) \prod_i P(a_i = w_k \mid y_j)$$

Estimating Likelihood

- Is problematic because we need to estimate it for each combination of text position, English word, and target value: 48*50,000*2≈5 million such terms.
- Assumption that reduced the number of terms – Bag of Words Model
  - The probability of encountering a specific word w_k is independent of the specific word position.

$$P(a_i = w_k \mid y_j) = P(a_m = w_k \mid y_j), \quad \forall i, m$$

  - Instead of estimating $P(a_1 = w_k \mid y_j),\ P(a_k = w_k \mid y_j), \ldots$ we estimate a single term $P(w_k \mid y_j)$
  - Now we have 50,000*2 distinct terms.

Estimating Likelihood

- The estimate for the likelihood is

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

n -the total number of word positions in all training examples whose target value is y_j

n_k -the number times word w_k is found among these n word positions.

$|Vocabulary|$ -the total number of distinct words found within the training data.

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* ← all word positions in *Doc* that contain tokens found in *Vocabulary*
- Return $y^* = \underset{y_j \in \{+,-\}}{\arg\max}\, P(v_j) \prod_{i \in positions} P(a_i \mid v_j)$