Classification-3

Dr. Parul Arora

Table of contents

- Probabilistic Generative model-Discrete i/ps
- Naive Bayes
- Example
- Summary

Probabilistic Generative Models – Discrete Inputs

- Consider first binary feature values $x_i \in \{0, 1\}$.

- Here we will make the naive Bayes assumption in which the **feature values are treated as independent**, conditioned on the class C_k. Thus we have class-conditional distributions of the form

$$p(x \mid C_k) = \prod_{i=1}^{D} \mu_{ki}^{x_i} (1 - \mu_{ki})^{1-x_i}$$

which contain D independent parameters for each class.

Probabilistic Generative Models – Discrete Inputs

- Substituting into the form needed for normalized exponential

$$a_k = \ln p(\mathrm{x}|\mathcal{C}_k)p(\mathcal{C}_k).$$

gives

$$a_k(\mathrm{x}) = \sum_{i=1}^{D} \{x_i \ln \mu_{ki} + (1 - x_i)\ln(1 - \mu_{ki})\} + \ln p(\mathcal{C}_k)$$

which again are linear functions of the input values x_i. For the case of K = 2 classes, we can alternatively consider the logistic sigmoid formulation.

Naïve Bayes

- Naive Bayes is an algorithm that falls under the domain of supervised machine learning, and relies on word frequency counts.
- Naive Bayes classifier assumes that the features are independent of each other. Since this is rarely possible in real-life data, so the classifier is called **naive**.
- This classification algorithm is based on **Bayes** theorem so known as Naive Bayes Classifier.
- Formally, the Naive Bayes inference condition rule for binary classification can be expressed as follows:

$$\prod_{i=1}^{m} \frac{P(w_i|pos)}{P(w_i|neg)}$$

- Naïve Bayes Algorithm is used in spam filtration, Sentimental analysis, classifying articles and many more.

Example: Corpus of movie reviews
Naive Bayesian Classifier
Features: adjectives (bag-of-words)

Doc	Words	Class
1	**Great** movie, **excellent** plot, **renowned** actors	Positive
2	I had not seen a **fantastic** plot like this in **good** 5 years. **Amazing**!!!	Positive
3	**Lovely** plot, **amazing** cast, somehow I am in love with the **bad** guy	Positive
4	**Bad** movie with **great** cast, but very **poor** plot and **unimaginative** ending	Negative
5	I **hate** this film, it has nothing **original.** Really **bad**	Negative
6	**Great** movie, but not...	Negative
7	Very **bad** movie, I have no words to express how I dislike it	Negative

Example: Corpus of movie reviews

- Convert the dataset in to frequency table
  - P(positive) = 3/7 = 0.43
  - P(negative) = 4/7 = 0.57

Example: Corpus of movie reviews

- Create Likelihood table by finding the probabilities.
- Likelihoods: Count adjectives x in class *C* /count of adjectives in *C*

• P(amazing | positive) = 2/9	P(amazing | negative) = 0/9
• P(bad | positive) = 1/9	P(bad | negative) = 3/9
• P(excellent | positive) = 1/9	P(excellent | negative) = 0/9
• P(fantastic | positive) = 1/9	P(fantastic | negative) = 0/9
• P(good | positive) = 1/9	P(good | negative) = 0/9
• P(great | positive) = 1/9	P(great | negative) = 2/9
• P(hate | positive) = 0/9	P(hate | negative) = 1/9
• P(lovely | positive) = 1/9	P(lovely | negative) = 0/9
• P(original I positive) = 0/9	P(original | negative) = 1/9
• P(poor | positive) = 0/9	P(poor | negative) = 1/9
• P(renowned | positive) = 1/9	P(renowned | negative) = 0/9
• P(unimaginative | positive) = 0/9	P(unimaginative | negative)= 1/9

Example: Corpus of movie reviews

- Use Naive Bayesian equation to calculate the posterior probability
- Given a new segment to classify (test time):

Doc	Words	Class
8	This was a fantastic story, good, lovely	???

- **Final decision**
- P(positive) P(fantastic |positive) P(good | positive) P(lovely | positive)

  (3/7) (1/9) (1/9) 1/(9) = 0.000059
- P(negative) P(fantastic | negative) P(good |negative) P(lovely |negative)

  (4/7) (0/9) (0/9) (0/9) = 0

- So: sentiment = positive

Example: Corpus of movie reviews

- Given a new segment to classify (test time):

Doc	Words	Class
9	Great plot, great cast, great everything	???

- **Final decision**
- P(positive) P(great |positive) P(great | positive) P(great | positive)

  (3/7) (1/9) (1/9) 1/(9) = 0.000059
- P(negative) P(great | negative) P(great |negative) P(great |negative)

  (4/7) (2/9) (2/9) (2/9) = 0.0063

- So: sentiment = negative

Example: Corpus of movie reviews

- Given a new segment to classify (test time):

Doc	Words	Class
10	Boring movie, annoying plot, unimaginative ending	???

- **Final decision**
- P(positive) P(boring |positive) P(annoying | positive) P(unimaginative| positive)

  (3/7) (0/9) (0/9) 0/(9) = 0
- P(negative) P(boring | negative) P(annoying |negative) P(unimaginative |negative)

  (4/7) (0/9) (0/9) (1/9) = 0

- So: sentiment = ???

Laplacian Smoothing

- **This is the problem of zero probability. So, how to deal with this problem?**
- Laplace Smoothing is a smoothing technique that handles the problem of zero probability in Naïve Bayes.

$$P(\mathrm{w_i}|\mathrm{class}) = \frac{\mathrm{freq}(\mathrm{w_i}, \mathrm{class})}{\mathrm{N_{class}}}$$

class ∈ {Positive, Negative}

$$P(\mathrm{w_i}|\mathrm{class}) = \frac{\mathrm{freq}(\mathrm{w_i}, \mathrm{class}) + 1}{\mathrm{N_{class}} + \mathrm{V}}$$

N_{class} = frequency of all words in class

V = number of unique words in vocabulary

$$P(W_{pos}) = \frac{freq(w_i, pos) + 1}{N_{pos} + V}$$

$$P(W_{neg}) = \frac{freq(w_i, neg) + 1}{N_{neg} + V}$$

Example: Corpus of movie reviews: Laplace Smoothing

- Add smoothing to feature counts (add 1 to every count) and number of distinct adjectives to all class counts.

Doc	Words	Class
10	Boring movie, annoying plot, unimaginative ending	???

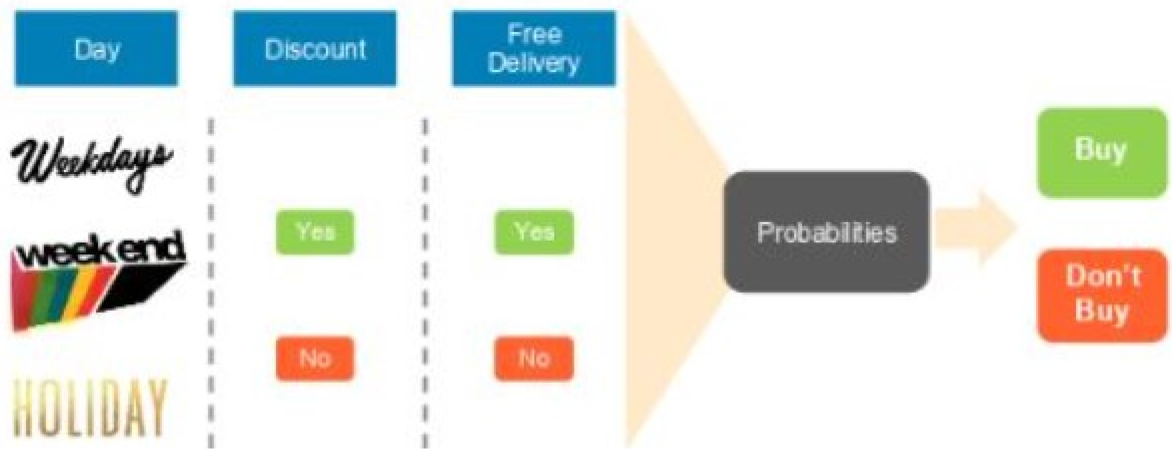
- **Final decision**
- P(positive) P(boring |positive) P(annoying | positive) P(unimaginative| positive)

  (3/7) ((0+1)/(9+12)) ((0+1)/(9+12)) ((0+1)/(9+12)) = 0.000046
- P(negative) P(boring | negative) P(annoying |negative) P(unimaginative |negative)

  (4/7) ((0+1)/(9+12)) ((0+1)/(9+12)) ((1+1)/(9+12)) = 0.000123
- So: sentiment = Negative

Shopping Example

Problem statement: To predict whether a person will purchase a product on a specific combination of day, discount, and free delivery using a Naive Bayes classifier.



Dataset:

Day	Discount	Free Delivery	Purchase
Weekday	Yes	Yes	Yes
Weekday	Yes	Yes	Yes
Weekday	No	No	No
Holiday	Yes	Yes	Yes
Weekend	Yes	Yes	Yes
Holiday	No	No	No
Weekend	Yes	No	Yes
Weekday	Yes	Yes	Yes
Weekend	Yes	Yes	Yes
Holiday	Yes	Yes	Yes
Holiday	No	Yes	Yes
Holiday	No	No	No
Weekend	Yes	Yes	Yes
Holiday	Yes	Yes	Yes
Holiday	Yes	Yes	Yes
Weekday	Yes	Yes	Yes
Holiday	No	Yes	Yes
Weekday	Yes	No	Yes
Weekend	No	No	Yes
Weekend	No	Yes	Yes
Weekday	Yes	Yes	Yes
Weekend	Yes	Yes	No
Holiday	No	Yes	Yes
Weekday	Yes	Yes	Yes
Holiday	No	No	No
Weekday	No	Yes	No
Weekday	Yes	Yes	Yes
Weekday	Yes	Yes	Yes
Holiday	Yes	Yes	Yes
Weekend	Yes	Yes	Yes

Step 1) Create frequency tables for each attribute using the input types mentioned in the dataset, such as days, discount, and free delivery.

Frequency Table		Buy	
		Yes	No
Discount	Yes	19	1
	No	5	5

Frequency Table		Buy	
		Yes	No
Free Delivery	Yes	21	2
	No	3	4

Frequency Table		Buy	
		Yes	No
Day	Weekday	9	2
	Weekend	7	1
	Holiday	8	3

Let the event ‘Buy’ denoted as ‘A’, and independent variables, namely ‘Discount’, ‘Free delivery’, and ‘Day’, denoted as ‘B’. We will use these events and variables to apply Bayes’ theorem.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Step 2) Now let us calculate the Likelihood tables one by one.

Frequency Table		Buy (A)		
		Yes	No	Row Sum
Day (B)	Weekday	9	2	11
	Weekend	7	1	8
	Holiday	8	3	11
Column Sum		24	6	

Frequency Table		Buy (A)		
		Yes	No	
Day (B)	Weekday	9/24	2/6	11/30
	Weekend	7/24	1/6	8/30
	Holiday	8/24	3/6	11/30
		24/30	6/30	

Step 2) Posterior probability calculation using Bayes theorem

- Based on this likelihood table, we will calculate the conditional probabilities as below.
  - P(A) = P(No Buy) = 6/30 = 0.2
  - P(B) = P(Weekday) = 11/30 = 0.37
  - P(B/A) = P(Weekday / No Buy) = 2/6 = 0.33

- Find P(A/B) using Bayes theorem,
  - P(A/B) = P(No Buy / Weekday) = P(Weekday / No Buy) * P(No Buy) / P(Weekday)
    = (2/6 * 6/30) / (11/30) = 0.1818

- Similarly, if A is Buy, then
  - P(A/B) = P(Buy / Weekday) = P(Weekday / Buy) * P(Buy) / P(Weekday) = (9/24 * 24/30) / (11/30) = 0.8181

As the P(Buy | Weekday) is more than P(No Buy | Weekday), we can conclude that a customer will most likely buy the product on a Weekday.

Similarly, we can calculate the likelihood of occurrence of an event on the basis of all the three variables.

- Calculate Likelihood tables for all three variables using frequency tables.

Frequency Table		Buy		
		Yes	No	
Discount	Yes	19/24	1/6	20/30
	No	5/24	5/6	10/30
		24/30	6/30	

Frequency Table		Buy		
		Yes	No	
Free Delivery	Yes	21/24	2/6	23/30
	No	3/24	4/6	7/30
		24/30	6/30	

- Using these likelihood tables, we will calculate whether a customer is likely to make a purchase based on a specific combination of ‘Day’, ‘Discount’ and ‘Free delivery’.
- Here, let us take a combination of these factors:
  - Day = Holiday
  - Discount = Yes
  - Free Delivery = Yes

- Calculate the conditional probability of purchase on the following combination of day, discount and free delivery.
- **When, A = Buy**
- Where B is:
  - Day = Holiday
  - Discount = Yes
  - Free Delivery = Yes
- = P(A/B)
- = P(Buy / Discount=Yes, Day=Holiday, Free Delivery=Yes)
- = (P(Discount=(Yes/Buy)) * P(Free Delivery=(Yes/Buy)) * P(Day=(Holiday/Buy)) * P(Buy)) / (P(Discount=Yes) * P(Free Delivery=Yes) * P(Day=Holiday))
- = (19/24 * 21/24 * 8/24 * 24/30) / (20/30 * 23/30 * 11/30)
- = 0.986

- Similarly, Calculate the conditional probability of purchase on the following combination of day, discount and free delivery.
- Where B is:
  - Day = Holiday
  - Discount = Yes
  - Free Delivery = Yes
- And A = No Buy
- P(A/B)
- = P(No Buy / Discount=Yes, Day=Holiday, Free Delivery=Yes)
- = (P(Discount=(Yes/No Buy)) * P(Free Delivery=(Yes/No Buy)) * P(Day=(Holiday/No Buy)) * P(No Buy)) / (P(Discount=Yes) * P(Free Delivery=Yes) * P(Day=Holiday))
- = (1/6 * 2/6 * 3/6 * 6/30) / (20/30 * 23/30 * 11/30)
- = 0.027
- We can conclude that the average customer will buy on a holiday with a discount and free delivery.

Probabilistic Generative Models - Summary

- For the two-class classification problem,

  - the posterior probability of class C_1 can be written as a *logistic sigmoid* acting on a linear function of x for a wide choice of class-conditional distributions $p(\mathrm{x}|C_k)$.

- Similarly, for the multiclass case

  - the posterior probability of class C_k is given by a *softmax transformation* of a linear function of x.

- For specific choices of the class-conditional densities $p(\mathrm{x}|C_k)$,

  -maximum likelihood (ML) to determine the parameters of the densities as well as the class priors $p(C_k)$ and then used Bayes' theorem to find the posterior class probabilities.

Thanks