Naive Bayes Classifier

Danushka Bollegala

Bayes Rule

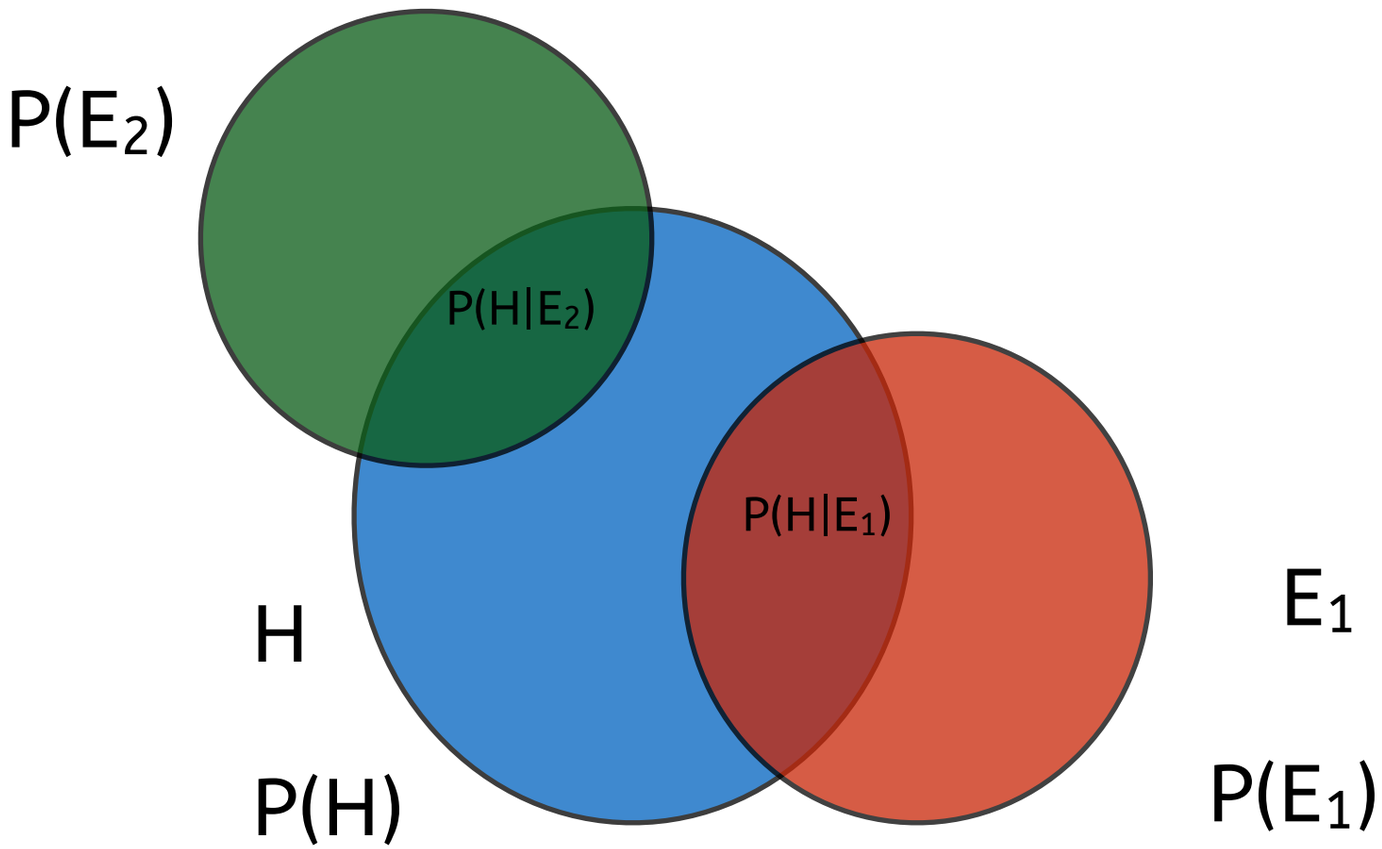
- The probability of hypothesis H, given evidence E
  - $P(H|E) = P(E|H)P(H)/P(E)$
- Terminology
  - P(E): Marginal probability of the evidence E
  - P(H): Prior probability of hypothesis H
  - P(E|H): Likelihood of the evidence given hypothesis
  - P(H|E): Posterior probability of the hypothesis H

Example

- Meningitis causes a stiff neck 50% of the time. Meningitis occurs 1/50000 and stiff neck occurs 1/20. Compute the probability of meningitis, given that the patient has a stiff neck?

- H = meningitis, E = stiff neck
- P(H) = 1/50000, P(E) = 1/20, P(E|H) = 0.5
- From Bayes' rule we have
  - P(H|E) = P(E|H)P(H)/P(E) = 0.0002

Intuition/Derivation



H can be related to several evidences E_1, E_2, …, E_n
$P(H) = P(E_1)P(H|E_1) + P(E_2)P(H|E_2)+\ldots+P(E_n)P(H|E_n)$

By the definition of conditional probability we have
$P(H|E) = P(H,E)/P(E)$
$P(E|H) = P(H,E)/P(H)$
Dividing one from the other we get
$P(H|E) = P(E|H)P(H)/P(E)$

Bayes Rule — Proportional Form

- Often the evidence is given (P(E) is fixed) and we need to select from a set of hypothesis $h_1, h_2, \ldots, h_k$
- In such cases we can simplify the formula to
  - $P(H|E) \propto P(E|H)P(H)$
  - posterior $\propto$ likelihood x prior
- At least remember this form!

Naive Bayes

- Let us assume a particular hypothesis H depends on several evidences $E_1, E_2, \ldots, E_n$
- From Bayes rule we have
  - $P(H|E_1, E_2, \ldots, E_n) \propto P(E_1, E_2, \ldots, E_n|H)P(H)$
- Let us further assume that given the hypothesis H, the evidences are **mutually exclusive**
  - Then we can decompose the likelihood term
  - $P(H|E_1, E_2, \ldots, E_n) \propto P(E_1|H) \ldots P(E_n|H)P(H)$
- This independence assumption is what make naive bayes so naive!

Independent Events

- Joint probability of independent events
  - P(A,B) = P(A|B)P(B) This holds for ANY two random events A and B, irrespective of whether they are independent or not.
    - But if A is independent of B, then B's occurrence has no consequence on A
    - P(A|B) = P(A)
- Therefore, when A and B are independent
  - P(A,B) = P(A)P(B)

Being naive makes life easy

- Let H=engine-does-not-start, and evidences A = weak-battery and B = no-gas
- P(H|A,B) = P(A,B|H)P(H)/P(A,B)
- We must estimate the likelihood P(A,B|H)
- If A and B are mutually independent given H
  - P(A,B|H) = P(A|H)P(B|H)
    - P(A|H) can be estimated by finding how many cars had engine not working because of a weak battery
    - P(B|H) can be estimated by finding how many cars had engines not working because of no gas
    - On the other hand, if we tried to estimate P(A,B|H) directly then we need to find how many cars had engines not working due to a weak battery and no gas. Such cases could be rare making our estimate of P(A,B|H) unreliable or zero (in the worst case).
- Making the independence assumption makes estimates possible in practice.

Predicting whether play=yes

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								
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								

To play or not to play

Given a test instance
x = (outlook=sunny, temp=cool, humidity=high, windy=TRUE)

P(play=yes|x) ∝ P(x|play=yes)P(play=yes)
= P(outlook=sunny|play=yes)xP(temp=cool|play=yes)x
P(humidity=high|play=yes)xP(windy=TRUE|play=yes) x P(play=yes)

=2/9 x 3/9 x 3/9 x 3/9 x 9/14 = 0.00529

P(play=no|x) ∝ P(x|play=no)P(play=no)
= P(outlook=sunny|play=no)xP(temp=cool|play=no)x
P(humidity=high|play=no)xP(windy=TRUE|play=no) x P(play=no)

=3/5 x 1/5 x 4/5 x 3/5 x 5/14 = 0.0020
Therefore play=yes.

Computing probabilities

- Note that
  - P(play=yes|x) ∝ 0.0052
  - P(play=no|x) ∝ 0.0020
- How can we compute the actual probabilities?
- Note that
  - P(play=yes|x) + P(play=no|x) = 1
- Therefore,
  - P(play=yes|x) = 0.0052 / (0.0052 + 0.0020) = 0.72

Sometimes it is too naive…

- Naive Bayes' assumption that the features are independent given the hypothesis is sometimes too naive to be true.
- The probability of Liverpool winning a football match is not independent of the probability for each member of the team scoring a goal.
- However, as we saw in a previous slide, it gives us a method to estimate the joint distribution of a set of random variables without getting into data sparseness issues.
- The linear classifiers we studied in the module so far such as the perceptron are also making such assumptions about the feature independence (the activation score is a linearly weighted sum after all)
- $\log(P(A,B|H) = \log(P(A|H) \times P(B|H)) = \log(P(A|H)) + \log(P(B|H))$

Zero probabilities

- Issue: If a feature value does not co-occur with a class value, then the probability generated for it will be 0.
- Eg. Given outlook=overcast, the probability of play=no is 0/5. The other features will be ignored as the final result will be multiplied by 0.
- This is bad for our 4 feature dataset, but terrible for (say) a 1000 feature dataset.
- In text classification, we often encounter situations where a feature does not occur in a particular class.

Laplace Smoothing

- We can “borrow” some probabilities from high probability features and distribute them among zero probability features to avoid having feature with zero probabilities
- This is called *smoothing*
- There are numerous smoothing techniques based on different policies. As long as the total probability mass remains unchanged any policy of probability reassignment is valid.
- A popular method is called **Laplace smoothing**
  - Add 1 to all counts to avoid zeros!
  - P(w) = (count(w) + 1) / (N + |V|)
  - count(w): the actual count (before smoothing) of a word w
  - N: corpus size in words (total number of counts of all words) $\sum_{w \in \mathcal{V}} \text{count}(w)$
  - |V|: vocabulary size (how many different words do we have)

Quiz

- Let count(cat)=9, count(dog)=0, and count(rabbit) = 1. Compute the probabilities P(cat), P(dog), and P(rabbit).

- Now smooth the above probabilities using Laplace smoothing.

Document Classification

- A document can be represented using a bag of words (features such as unigrams and bigrams). We could represent a document by a vector where each element corresponds to the total frequency of a feature in the document.
- D = “the burger i ate was an awesome burger”
- v(D) = {the:1, burger:2, i:1, ate:1, was:1, an:1, awesome:1}

- Assuming the features to be independent (the *naive* assumption) we can compute the likelihood (probability) of this document D, p(D) as follows
- $p(D) = p(the)^{1} p(burger)^{2} p(ate)^{1} p(was)^{1} p(an)^{1} p(awesome)^{1}$
- If a word w occurs n times in D, then the term corresponding to p(w) appears n times in the product. Therefore, we have $p(w)^{n}$ in the likelihood computation above.

Document Classification

- The Bayesian model is often used to classify documents as it deals well with a huge number of features simultaneously.
- But we might know how many times a word occurs in the document (vector representation in the previous slide)
- This leads to Multinomial Naive Bayes
- Assumptions:
  - Probability of a word occurring in a document is independent of its location within the document.
  - The document length is not related to the class.

Multinomial Distribution

- This is an extension of the Binomial distribution for more than two classes
- Binomial distribution
  - What is the probability that when I flip a coin n times I will get k number of heads (H) and (n-k) number of tails (T)?

$$p(H = k, T = n - k) = \frac{n!}{k!(n-k)!} p^k (1-p)^{(n-k)}$$

- Multinomial distribution
  - Lets flip a dice instead of a coin. There are six outcomes.

$$p(1 = a, 2 = b, \ldots, 6 = f) = \frac{n!}{a!b!\ldots f!} p(1)^a p(2)^b \ldots p(6)^f$$

$$n = a + b + c + d + e + f$$

Document Classification

$$P(\boldsymbol{x}|y) = N! \prod_{w \in D} \frac{p(w|y)^{h(w,D)}}{h(w,D)!}$$

- N = number of words in the document **x**
- p(w|y) = probability that the word w occurs in class y
- h(w,D) = total occurrences of the word w in document D

Classifying “burger” sentiment

- Let us assume that we would like to classify whether the following document is positive (1) or negative (-1) in sentiment.
- D = “the burger i ate was an awesome burger”
- Further assume we see these words in positive and negative classes as follows. To make our computations easier let us assume that we removed “the”, “i”, “was”, “an” as stop words.

word	+1	-1
burger	3	2
ate	3	2
awesome	4	1

Computing class conditional probabilities

word	+1	-1	p(w|+1)	p(w|-1)
burger	3	2	3/10	2/5
ate	3	2	3/10	2/5
awesome	4	1	4/10	1/5

Quiz

- Using the probabilities in the table in slide 21 and multinomial naive Bayes formula in slide 19, compute P(D|+1) and P(D|-1)

Quiz

- Assuming that both positive and negative classes have equal probability
  - p(+1) = p(-1) = 0.5
- Compute P(+1 | D) and P(-1 | D) for the document in slide 22.
- Is this document D positive or negative?