Finish kNN, Introduction to Naive Bayes Algorithm

Jaya Nilakantan

Dawson College

jnilakantan@dawsoncollege.qc.ca

Recap

We've seen 2 algorithms thus far:

- Linear regression: used to predict a value based on a single feature. Assumes a continuous linear relationship
- k-nearest neighbours: used to predict a classification based on feature(s).

Both algorithms are used with supervised learning: assumes labeled data

How to choose k in kNN?

We saw that the kNN algorithm is sensitive to the value of k
Different results can occur depending on:

- the distribution of the test data points
- the value of k

In other words, the accuracy of the results depends on the value of k. We need to *tune* k to acheive the optimal results (i.e., lowest error rates)

Optimizing k

There isn't a general formula to determine the best value of k.
Instead, you need to tune k for each data set that you are examining.
We need to tune k in the same way that we tune the weights w_i
The common methodology used is called *k-fold cross validation*

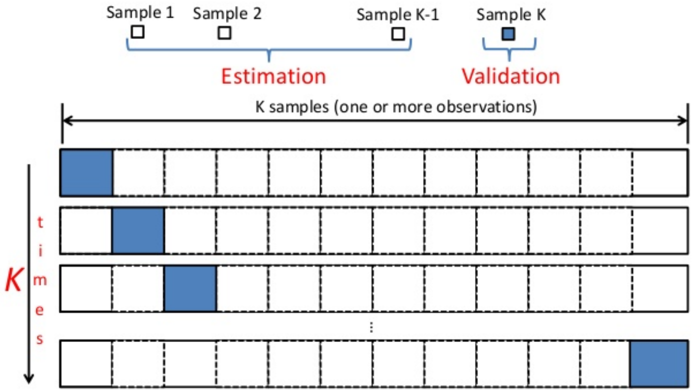
k-fold cross validation

- split your training data into a bunch of random, equal sized groups (called *folds*, there are *k* of them. Beware, *k* refers to a different variable than the *k* in kNN!)
- use one of the folds as the "validation" data, and the remaining folds as the "training" data: run through different values of k in kNN, and assess the accuracy
- repeat this for every fold (i.e., each fold takes a turn being the validation data)

Visualization

Cross-validation: How it works?

- K-fold cross-validation:



Pseudo code

```
Initialize value of k-fold
Split test data into k-fold folds
for each fold
  for loop to iterate through different values for kNN's k
    the iterating fold is the validation fold, all remainin
    use the data in the validation fold to count the numbe
  save the accuracy rate for that k
select the k with the best accuracy
```

How would you use kNN for text classification?

1. Transform the text into numeric vectors
2. use a distance metric to calculate the closest neighbours

kNN classification assumes that similarity can be found be proximity. How can the vectors be formed so that proximity is meaningful?

Naive Bayes classifiers

Naive Bayes classifiers are probabilistic classifiers
Bayes refers to Bayes theorem which defines a relation between conditional probabilities.
Naive refers to the fact that the features are independant (which is a big assumption)

Bayes Theorem

This video has an easy to understand explanation.

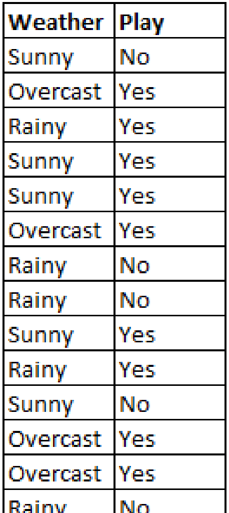
Time for math!

The Bayes theorem states

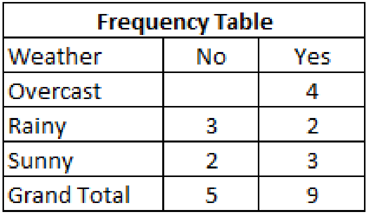
$$P(A \mid B) = \frac{P(B \mid A)\, P(A)}{P(B)}$$

Simple example 1

Based on the weather (feature), the kids either played outside or not (class). We have collected past data, and want to predict the likelihood of them playing outside given the weather.



Weather	Play
Sunny	No
Overcast	Yes
Rainy	Yes
Sunny	Yes
Sunny	Yes
Overcast	Yes
Rainy	No
Rainy	No
Sunny	Yes
Rainy	Yes
Sunny	No
Overcast	Yes
Overcast	Yes
Rainy	No



Frequency Table		
Weather	No	Yes
Overcast		4
Rainy	3	2
Sunny	2	3
Grand Total	5	9

Continuing example 1

If it is sunny, will the kids play outside?

$$\begin{aligned} P(Yes|Sunny) &= \frac{P(Sunny|Yes) * P(Yes)}{P(Sunny)} \\ &= ((3/9) * (9/14))/(5/14) = 0.6 \\ P(No|Sunny) &= \frac{P(Sunny|No) * P(No)}{P(Sunny)} \\ &= ((2/5) * (5/14))/(5/14) = 0.4 \end{aligned}$$

So the children are more likely to play outside.

Application to text classification?

We want to classify e-mails into classes c (e.g., spam, not spam)
There are features x_i (the words) that define e-mails
We know the unconditional probabilities $P(c)$ for each class

- this is simply the frequency with which spam and not-spam appears in our training set

We know the conditional probability $P(x_i \mid c)$ for each feature and class

- this is the frequency of the occurrence of the word in each class
- this is a measure of how much evidence that particular feature contributes to c being the correct class

Calculation of $P(c \mid x)$

We want to calculate the probability that a given email is spam. We are interested in:

The calculation of the conditional probability that the class is c, given the words x_i in the email:

$$P(c \mid x_1, x_2, ...x_n) = \frac{P(x_1, x_2, ...x_n \mid c)\, P(c)}{P(x_1, x_2, ...x_n)}$$

Notice the denominator doesn't depend on the class, and is effectively constant when I am comparing if a text belongs to one class or another. So we can ignore it since it doesn't change which class will have the higher **relative** probability

Now consider the naive part

The Naive Bayes model, which we are using, assumes every feature is independant of each other.
In other words, it assumes x_i are all independant, meaning their probabilities don't impact each other
This means that:

Since the features x_i are assumed to be independant

$$P(x_1, ... x_n \mid c) = P(x_1|c)\ P(x_2|c) \quad P(x_n|c)$$

So we can rewrite the Naive Bayes equation:

$$\boxed{P(c \mid x_1, ...x_n) = P(x_1|c)\ P(x_2|c)\quad P(x_n|c)P(c)}$$

Note 1: We removed the denominator, which was less that 1 in most cases. This will make our probability estimate smaller than it really is. We don't mind since we are comparing between 2 option for the higher *relative* value
Note 2: if the probability of any one term being in the class is zero, the entire term will multiply to 0. To protect us against this, we will use something called *Laplace smoothing*, which adds one.

Let's look at an example

Consider this training set:

DocID	words	class
1	Chinese Beijing Chinese	China
2	Chinese Chinese Shanghai	China
3	Chinese Macao	China
4	Tokyo Japan Chinese	not China
5	Chinese Chinese Chinese Tokyo Japan	?

Document 1 to 4 are out training set. The words are the features x, and the class c is China or Japan.
How would we classify Document 5? China or Not China? Which has a higher probability?

Applying Naive Bayes

Remember:

$$\boxed{P(c \mid x_1, \ldots x_n) = P(x_1|c)\ P(x_2|c)\quad P(x_n|c)P(c)}$$

or

```
 P(China | "Chinese Chinese Chinese Tokyo Japan" )
   = P("Chinese"|China) P("Chinese"|China) P("Chinese"|Chin
   P("Tokyo"|China) P("Japan"|China)  P(China)

P(China) = 3/4 since 3/4 training documents are classified
P("Chinese"|China) = (5 occurrences of "Chinese" associate
  plus 1 for Laplace smoothing)/
  (8 terms associated with class China + overall vocabular
  = 6/14
P("Tokyo" | China) = 0 occurrences of "Tokyo" associated w
  plus 1 for Laplace smoothing)/
  (8 terms associated with class China + overall vocabular
  = 1/14
```

China or Not China

P(China | Doc 5) $= (6/14)^3 * 1/14 * 1/14 * 3/4 \sim=$ 0.0003
You can verify on your own:
P(Not China | Doc 5) $= (2/9)^3 * 2/9 * 2/9 * 1/4 \sim=$ 0.0001
Thus the Naive Bayes classifier will assign “China” as the class for the text since the probability is higher

Working with a real example

NY Times offers a wonderful set of APIs to beautiful labelled data. Get an API key, and play around. I am using data extracted previously (by Hilary Mason) that has 47 article snippets related to Arts, and 47 related to Sports.
You will use these as training data for a Naive Bayes classifier, which takes a sentence and decides if it is more likely to be Arts-related or Sports-related

NB Pseudo code

$P(Arts) = P(Sports) = 0.5$ since we chose equal number of articles. If this is not a good assumption, we should rethink our training data.
Since they are the same, we can ignore it in our relative probability calculations.

Training

Separate the text into words
Ignore words like “the” and “a”
Fill two associate arrays, one for arts, one for sport.

- key is a word, value is the frequency
- PHP arrays are hashmaps internally, so we will have O(1) access time later
- no need to sort

Count the number of words associated with Arts: sum of all values
Count the number of words associated with Sports: sum of all values
Count the overall number of unique words in the training set: count of all keys

Testing

Separate the sentence into words
Sum the frequency of the words in Arts plus 1
Sum the frequency of the words in Sports plus 1
Use these to calculate which has a higher probability

Using PHP-ML

The PHP Machine Learning library comes with a NaiveBayes Classifier class
You provide the training data as an array: so you will have to split the sentences, with a matching labels array. Training can be done in multiple steps so use a loop.
Once the system is trained, you can predict with an array of the features of the sentence.

Remember the assumptions with Naive Bayes

1. each feature is independant of the other
2. the probability of a future conditional event is **estimated** by the frequency of past events

*

References

https://kevinzakka.github.io/2016/07/13/k-nearest-neighbor/#more-on-k

https://en.wikipedia.org/wiki/Bag-of-words_model

https://nlp.stanford.edu/IR-book/html/htmledition/naive-bayes-text-classification-1.html

https://srdas.github.io/MLBook/BayesModels.html#bayes-classifier

https://en.wikipedia.org/wiki/Law_of_total_probability

https://en.wikipedia.org/wiki/Naive_Bayes_classifier

https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/

https://github.com/hmason/ml_class/tree/master/intro_web_data