Computational Data Science

Joel Huang

September 25, 2018

1 Entity Relationship Diagrams & SQL

2 Lab: Data Handling in Unix

Previewing data

Raw data can be fairly large. We might want to examine the dataset's structure without opening it in a special program or text editor, which could take a long time and large amounts of memory.

Preview all - `cat`

Short for con**cat**enate. Sequentially reads file(s) and writes them to `stdout`. If redirection `>` is used, then output is written to the specified file. `>` is used to write to a file and `>>` is used to append to a file.

```
cat file1.txt
cat file1.txt file2.txt > newcombinedfile.txt
cat >newfile.txt
cat -n file1.txt file2.txt > newnumberedfile.txt
cat file1.txt >> file2.txt
cat file1.txt file2.txt file3.txt | sort > test4
```

Preview some - `head, tail`

Use `head` or `tail` to preview the head or tail of the data. Remember to supply the flags:

```
-n  Number of lines
-B  Display number of lines before
-A  Display number of lines after
```

Searching - `grep`

```
-E  Use extended regular expression syntax
-o  Output a matching segment of each line only
-n  Print the line number of each matched line
-C  Show a number of context lines too
```

```
.   Matches any character.
*   Matches zero or more instances of the
     preceding character.
+   Matches one or more instances of the
     preceding character.
[]  Matches any of the characters within the
     brackets.
()  Creates a sub-expression that can be combined
     to make more complicated expressions.
|   OR operator; (www|ftp) matches either 'www'
     or 'ftp'.
^   Matches the beginning of a line.
$   Matches the end of the line.
\   Escapes the following character. Since .
     matches any character, to match a literal
     period you would need to use \..
```

3 Big Data, Hadoop, & MapReduce

4 Lab: MapReduce & Hadoop

Big Idea: Mappers & Reducers

5 Counting Relative Frequencies

Big Idea: Approximate probabilities by counting occurrences in the data

k-Nearest Neighbours

A way to measure similarity between different data points, through determining similarity values or distances. Classification with k-NN is straightforward, but sensitive to the value of k, potentially computationally expensive (but can be sped up using kd-tree), and takes memory to store all data points.

Decision trees & Random Forests

Each feature represents an opportunity for branching. Decision trees are inexpensive and explainable, but prone overfitting withouot pruning To prevent overfitting, use Random forests-ensemble approach that aggregates decision trees's outputs via a majority voting system

Naive Bayes for classification

```
Assumption: Conditional independence
```

$$P(x_1, x_2, \ldots, x_n \mid Y) = P(x_1 \mid Y) \cdot P(x_2 \mid Y) \cdot \ldots \quad (1)$$

We are trying to find the most likely class given an observation Maximum Most likely class Y given feature set X

$$y_{MAP} = \arg\max \; P(Y \mid x_1, x_2, \ldots, x_m) \quad (2)$$

$$= \arg\max \; \frac{P(X \mid Y)\, P(Y)}{P(X)} \quad (3)$$

In equation (2), we're comparing the $\arg\max$ of $P(Y = 0 \mid X)$ and $P(Y = 1 \mid X)$. Therefore we can cancel the constant denominator $P(X)$:

$$= \arg\max \; P(X \mid Y)\, P(Y) \quad (4)$$

Next, we need to find $P(X \mid Y)$ and $P(Y)$.

Count the prior probability $P(Y)$ Class labels are $Y \in \{0, 1\}$. Therefore, for a dataset with n labels,

$$P(Y = 0) = \frac{n_{Y=0}}{n} \quad (5)$$

$$P(Y = 1) = \frac{n_{Y=1}}{n} \quad (6)$$

In other words, there are $n_{Y=1}$ out of n occurrences of label Y having value 1.

Estimate $P(X \mid Y = 0)$ **and** $P(X \mid Y = 1)$ Since we have assumed conditional independence of X given classes Y (1),

$$P(x_i \mid Y = 0) = \frac{|x_{i,Y=0}|}{n_{Y=0}} \quad (7)$$

We are counting the probability (occurrences in the dataset) of feature x_i given $Y = 0$. Find all the rows where $Y = 0$ and count the occurrences of feature x_i. Repeat this for $Y = 1$.

Final comparison ($\arg\max$) Now we have pairs $P(Y = 0)$, $P(X \mid Y = 0)$, and $P(Y = 1)$, $P(X \mid Y = 1)$. Multiply the pairs as in equation (4) and take the *max* of the two.

Naive Bayes for word counting

```
Assumptions: Conditional independence,
    irrelevance of word order
```

We want to predict a class for a given sentence, for example, "good" for the sentence "love the burgers here, delicious and filling". We can estimate the class c using Naive Bayes,

$$c = \arg\max_{c_i \in C} P(c_i) \prod_{w_j \in W} P(w_j \mid c_i) \quad (8)$$

For a set of documents D, words W, and classes C, the probability of class c_i is simply the number of occurrences of c_i in document D.

$$P(c_i) = \frac{|c_i|}{|D|} \quad (9)$$

The probability of word w_1 given class c_i is also simply the number of occurrences of word w_1 in sentences with class c_i, divided by the probability of class c_i.

$$\begin{aligned} P(w_1 \mid c_i) &= \frac{P(w_1, c_i)}{P(c_i)} \\ &= \frac{count(w_1, c_i)}{\sum_{w_j \in W} count(w_j, c_i)} \end{aligned} \quad (10)$$

Example: Restaurant ratings

ID	Sentence	Class
1	The burger is **tasteless** and **slow** service	Bad
2	**slow** serving time and everything is **horrible**	Bad
3	Restaurant is near MRT, serves **delicious** burgers	Good
4	**love** the burger here, **delicious** and filling	Good
5	**love** this place, **delicious** burgers but **slow** service	?

Where classes $C = \{Bad, Good\}$, and words $W = \{tasteless, slow, horrible, delicious, love\}$.

$$P(c) = \begin{cases} \frac{2}{4}, & c = Bad \\ \frac{2}{4}, & c = Good \end{cases} \quad (11)$$

$$\begin{aligned} P(love \mid Good) &= \frac{count(love\,, Good)}{\sum_{w_j \in W} count(w_j\,, Good)} = \frac{1}{3} \\ P(delicious \mid Good) &= \frac{count(delicious\,, Good)}{\sum_{w_j \in W} count(w_j\,, Good)} = \frac{2}{3} \end{aligned} \quad (12)$$

*** TODO: CHECK THIS**

$$
\begin{aligned}
P(Good \,|\, love, delicious) &= P(Good) \prod_{w_j \in W} P(w_j \,|\, Good) \\
&= P(Good) \cdot P(love \,|\, Good) \cdot P(delicious \,|\, Good) \\
&= \frac{2}{4} \times \frac{1}{3} \times \frac{2}{3} = \frac{1}{9}
\end{aligned} \tag{13}
$$

$$
\begin{aligned}
P(tasteless \,|\, Bad) &= \frac{count(tasteless\,,\, Bad)}{\sum_{w_j \in W} count(w_j\,,\, Bad)} = \frac{1}{4} \\
P(slow \,|\, Bad) &= \frac{count(slow\,,\, Bad)}{\sum_{w_j \in W} count(w_j\,,\, Bad)} = \frac{2}{4} \\
P(horrible \,|\, Bad) &= \frac{count(horrible\,,\, Bad)}{\sum_{w_j \in W} count(w_j\,,\, Bad)} = \frac{1}{4}
\end{aligned} \tag{14}
$$

*** TODO: INCOMPLETE**

$$
\begin{aligned}
P(Bad \,|\, love, delicious) &= P(Bad) \prod_{w_j \in W} P(w_j \,|\, Bad) \\
&= P(Bad) \cdot P(love \,|\, Bad) \cdot P(delicious \,|\, Bad) \\
&= \frac{2}{4} \times \frac{1}{3} \times \frac{2}{3} = \frac{1}{9}
\end{aligned} \tag{15}
$$