Sample questions with solutions

Ekaterina Kochmar

May 27, 2017

Question 1

Suppose there is a movie rating website where User$_1$, User$_2$ and User$_3$ post their reviews. User$_1$ has written 30 positive (5-star reviews), 40 neutral (2 to 4 stars), and 30 negative (1-star) reviews in total; User$_2$ has written 10 positive, 10 neutral, and 0 negative reviews; and User$_3$ has written 30 positive, 30 neutral and 40 negative reviews.

The website uses a system that selects one of the reviews from this set to post on the main page every day.

(a) If a review posted today is chosen by such a system at random with probabilities $p(User_1) = 0.2$, $p(User_2) = 0.2$, and $p(User_3) = 0.6$, and then a review is selected at random (with equal probability of selecting any of the reviews from the chosen user), what is the probability that a *positive* review is posted on the main page?

(b) If we observe that the review posted on the main page today is in fact *neutral*, what is the probability that it was written by $User_1$?

Solutions

This question is testing basic understanding of probabilities and is an adapted version of question 1.3 from Chris Bishop's "Pattern Recognition and Machine Learning" book: the original question uses boxes of different colour and fruits drawn from boxes, but I made it more topically relevant for the course.

(a) The following are the probabilities of randomly choosing a review with a certain sentiment from each of the users:

	pos	neut	neg	Total
#	30	40	30	100
p	0.3	0.4	0.3	1.0

Table 1: User$_1$

	pos	neut	neg	Total
#	10	10	0	20
p	0.5	0.5	0.0	1.0

Table 2: User$_2$

	pos	neut	neg	Total
#	30	30	40	100
p	0.3	0.3	0.4	1.0

Table 3: User$_3$

$\mathbf{p(pos)} = p(User_1) * p(pos|User_1) + p(User_2) * p(pos|User_2) + p(User_3) * p(pos|User_3)$
$= 0.2 * 0.3 + 0.2 * 0.5 + 0.6 * 0.3 = 0.06 + 0.10 + 0.18 = \mathbf{0.34}$

$p(neut) = p(User_1) * p(neut|User_1) + p(User_2) * p(neut|User_2) + p(User_3) * p(neut|User_3)$
$= 0.2 * 0.4 + 0.2 * 0.5 + 0.6 * 0.3 = 0.08 + 0.10 + 0.18 = 0.36$

$p(neg) = p(User_1) * p(neg|User_1) + p(User_2) * p(neg|User_2) + p(User_3) * p(neg|User_3)$
$= 0.2 * 0.3 + 0.2 * 0.0 + 0.6 * 0.4 = 0.06 + 0.00 + 0.24 = 0.30$

(b)

$\mathbf{p(User_1|neut)} = \frac{p(User_1 \cap neut)}{p(neut)} = \frac{p(neut|User_1)*p(User_1)}{p(neut)} = \frac{0.4*0.2}{0.36} = \mathbf{2/9}$
$p(User_2|neut) = \frac{p(User_2 \cap neut)}{p(neut)} = \frac{0.5*0.2}{0.36} = 5/18$
$p(User_3|neut) = \frac{p(User_3 \cap neut)}{p(neut)} = \frac{0.3*0.6}{0.36} = 1/2$

Question 2

This question is on analysis and interpretation of the results of ML algorithms.

Suppose your inbox contains 100 emails, 19 of which are spam and the rest are emails from your friends, University, etc. (non-spam). You try out two spam filters:

Filter1:

	Non-spam (predicted)	Spam (predicted)	Total
Non-spam (actual)	65	16	81
Spam (actual)	4	15	19
Total	69	31	100

Filter2:

	Non-spam (predicted)	Spam (predicted)	Total
Non-spam (actual)	80	1	81
Spam (actual)	15	4	19
Total	95	5	100

What is the performance of each of the filters? If you only had a choice between these two filters in practice, which one would you choose and why?

Solutions

Table below presents the results.

	Acc	P(s)	R(s)	F_1(s)	P($\overline{s}$)	R($\overline{s}$)	F_1($\overline{s}$)
Filter1	0.80	0.48	0.79	0.60	0.94	0.80	0.86
Filter2	0.84	0.80	0.21	0.33	0.84	0.99	0.91

Table 4: Performance of the two spam filters in terms of accuracy (Acc), precision (P), recall (R) and F_1 on spam (s) and non-spam ($\overline{s}$).

The following observations can be made:

- At first glance, the two filters perform reasonably well: the accuracy of Filter1 is 0.80 and that of Filter2 is 0.84. However, it is important to keep in mind that the baseline – the majority class distribution, or the accuracy the filter would achieve if it simply identified all emails as non-spam – is 0.81. This suggests that Filter1, in fact, performs poorly. Even though performance of Filter2 is only 3 points higher than the baseline, it outperforms both the baseline and Filter1.
- The main problem with Filter1 is that it misidentifies 16 non-spam emails as spam. That is, more than half of the emails in spam folder are in fact not spam emails: precision on spam class is lower than 0.5 (0.48).
- Filter2, in contrast, misses many spam emails and puts them in the inbox (note low recall of 0.21 on spam). The advantage of using Filter2 is that it has high precision in identifying spam: as compared to Filter1 which puts many non-spam emails in spam folder, it only misidentifies 1 non-spam email as spam.
- To summarise, Filter1 has higher recall and F_1 on spam class and higher precision on non-spam class; Filter2 has overall higher accuracy, higher precision on spam, and higher recall and F_1 on non-spam. Since none of the two filters is flawless, which one should be preferred?
  The decision in this case should be based on the type of the application. For example, in spam filtering the cost of false positives (normal emails misidentified as spam and sent to spam folder) is higher than the cost of false negatives (spam emails that are kept in inbox). The latter might be annoying but the former still cause more serious problems: imagine not receiving an important email because it was sent to spam folder.
- In many applications more weight should be put on precision than on recall: credit card fraud detection and machine learning-based medical diagnosis applications are good examples. F_1 puts equal weight on precision and recall, but can be adapted to emphasise precision (e.g., $F_{0.5}$) or recall (e.g., F_2).

Question 3

Questions on statistical laws of language (bookwork)

Briefly summarise Heaps' law and Zipf's law. What each of these tell you about the distribution of words in language? Why and how this can be taken into account in practical applications?

Solutions

Based on §5.1 of `https://nlp.stanford.edu/IR-book/pdf/05comp.pdf`. Slight modifications to this question and solution by AC.

Heaps' law:

- Heaps' law relates vocabulary size $|V|$ to the number of tokens T in a text collection. It uses two experimentally determined parameters k and b (typically about $30 \leq k \leq 100$ and $b \approx 0.5$ for English).

$$|V| = kT^b$$

- Thus the relationship between collection size and vocabulary size is (roughly) linear in log-log space.
- The parameter k is very variable because vocabulary growth depends a lot on the nature of the collection and how it is processed: case-folding and stemming reduce the growth rate of the vocabulary, whereas including numbers and spelling errors increase it.
- Heaps' law demonstrates that (1) the dictionary size continues to increase with more documents in the collection, rather than a maximum vocabulary size being reached, and (2) the size of the dictionary is quite large for large collections.
- This demonstrates that with any NLP application, there will be previously unseen words. We thus have to allow for these: e.g., with smoothing techniques in machine learning.
- Once the parameters are determined, Heaps' Law can be used to estimate vocabulary size for a new text collection: this is useful for designing systems and experiments.
- We can also conclude that dictionary compression can significantly help speeding up text processing in large collections, which is important in many practical applications, e.g. in information retrieval systems.

Zipf's law:

- Zipf's law states that if t_1 is the most common term in the collection (rank 1), t_2 is the next most common (rank 2), and so on, then the collection frequency cf_i of the i-th most common term (rank i) is proportional to $1/i$: i.e., $cf_i \propto \frac{1}{i}$
- The law shows that if the most frequent term occurs cf_1 times, then the second most frequent term has half as many occurrences, the third most frequent term a third as many occurrences, and so on. The intuition is that frequency decreases very rapidly with rank.
- Alternatively, Zipf's law can be written as $cf_i = ci^k$ or as $log\ cf_i = log\ c + k\ log\ i$ where $k = 1$ and c is a constant. It is therefore a *power law* with exponent $k = -1$.
- What Zipf's law suggests for machine learning is that we will sample a lot of the high frequency items (words, but also phrases etc etc) with a relatively small amount of training data. It also reinforces the point about smoothing made above with respect to Heaps' Law.

Explanatory note: it's clear that Heaps' Law is related to Zipf's Law, and various authors have suggested that Heaps' Law can be derived from Zipf's Law, but the exact nature of the relationship is still a research topic. There are other related laws.

Question 4

(a) Describe an example of a graph where the diameter is more than three times as large as the average distance.

(b) Describe the concept of a triadic closure in social network theory.

Solutions

Questions on graphs from 2.5 (p. 44) or 3.7 (p. 83) of the course book (`https://www.cs.cornell.edu/home/kleinber/networks-book/networks-book.pdf`*).*

(a) *Diameter* of a graph is the maximum distance between any pair of nodes in the graph. *Average distance* is the average distance over all pairs of nodes in the graph.

A graph with fully connected nodes (e.g., a clique) will have an average distance of 1. If we connect one more node to one of the existing nodes, such that the distance to it is 3, the diameter will be more than three times as large as the average distance.

Social (as well as collaboration) networks represent a realistic example of such a graph: we can imagine a circle of friends closely connected to each other and having an average distance which is only slightly over 1, with a few nodes being separated by much larger distance and thus contributing to larger diameter of the graph.

(b) *Bookwork.* Triadic closure is discussed in detail in §3.1 of `https://www.cs.cornell.edu/home/kleinber/networks-book/networks-book.pdf`.

Triadic closure principle: If two people in a social network have a friend in common, then there is an increased likelihood that they will become friends themselves at some point in the future.

I.e., if nodes B and C have a friend A in common, then the formation of an edge between B and C produces a situation in which all three nodes A, B, and C have edges connecting each other – a structure we refer to as a *triangle* in the network. The term "*triadic closure*" comes from the fact that the $B - C$ edge has the effect of "closing" the third side of this triangle. If we observe snapshots of a social network at two distinct points in time, then in the later snapshot, we generally find a significant number of new edges that have formed through this triangle-closing operation, between two people who had a common neighbour in the earlier snapshot.