Lecture 4 - Good-Turing probability estimation.*

http://kochanski.org/gpk, http://www.staff.uiuc.edu/~cls/

March 6, 2006

1 Probabilities of Probabilities

If we're not careful, we will end up talking about probabilities of probabilities, using sentences like "...the estimator gives the value of the underlying probability that is most probable..." That sentence makes sense and is correct, but keeping the two different probabilities apart can put a bit of strain on one's cranium.

To reduce the cranial strain, we will change the terminology a bit. Instead of trying to estimate the probability of observing some word $P(W)$, we'll imagine that we have an huge (infinite) corpus and talk about the frequency at which we observe the word, F_W. We'll call this hypothetical frequency observed in a near-infinite corpus the "underlying frequency" or sometimes just the frequency. Now, $F_W = P(W)$: if we have a large enough corpus so that we see each possible English sentence many times[1], then the rate of observing each word or N-gram or sentence is just equal to the probability. So we're really talking about the same thing; we've changed nothing but the name. Still, this lets us talk about ideas like "The frequency is 90% likely to between 0.0012 and 0.0013" with less confusion.

The obvious thing we can do now is to use Bayes' Theorem and compute $P(F_W|D)$ for all possible values of the frequency of the word under considera-

[1]...and that's a rather large corpus. If we assume there are about 100,000 words in English and that the longest sentence we care about is 30 words, then there are $100,000^{30}$ possible sentences. If we write that out without exponential notation, there are 1,000,000, 000,000,000,000,000,000,000,00,000,000,000,000,000,000,000,000,000,000,000,000,000,000, 000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000, 000,000,000 possible sentences, a number which is comfortably larger than the number of atoms in the observable universe. Of course, some of those sentences will occur much more often than others, so if we want many examples of the rare sentences, we'll have to add a few zeros onto the size of the corpus. Actually collecting the corpus will be left as an exercise to the reader.

tion[2]. Then, we could take the most probable model, $\hat{F}$ and use it to predict things, build it into a speech recognition system or whatever. This would be the maximum likelihood value for F_W. This is a common choice, but not the only one, and we'll find some important examples where it is a decidedly bad idea.

2 Maximum Likelihood estimation

The simplest way to estimate frequencies of things you count is by dividing the counts for one item by the total counts. For example, if you have a corpus of words, and the word "the" appears 570 times out of 10000 words, this simple estimate of F_{the} is 0.057.

This ratio approaches the underlying probability as you take more samples. In the limit of an infinite number of samples, it gives you the probabilities exactly, but for any finite corpus, all you can say is that the maximum likelihood estimator is near the underlying frequency.

The frequency: The phrase "the frequency" hides the assumption that the data is generated by a single random process that doesn't change with time or depend on external events. If those conditions aren't met, you need to be careful to precisely define which probability you are talking about.
For instance, rather than saying "the frequency of observing spiders", one might have to say "the average frequency of observing spiders, over an entire year", because you will find more spiders in some seasons than in others.

This works in a full corpus or in any selected subset. All you have to do is count in the universe specified by the condition, and keep the total of events in the smaller universe. For instance $F(\text{the}|\text{Shakespeare})$ can be had by counting the word "the" in a corpus of Shakespeare's writing, and then counting the total number of words Shakespeare wrote.

3 Another way to estimate probabilities

The trouble with the maximum likelihood estimator (MLE) is that it predicts that the probability of a word that you haven't seen is exactly zero. That's fine for some applications, but not for others.

The trouble can come when you are trying to use counts of one corpus to estimate what you will observe in another corpus. Modern newspaper stories, for instance might have zero occurrences of "methinks", so $F(\text{methinks}) = 0$. If you use that as a model of English, the model will predict with certainty that

[2] E.g. 0.00000000000000000001, 0.00000000000000000010001, 0.00000000000000000010002, 0.00000000000000000010003, ... 0.00000000000000000010113, 0.00000000000000000010114, ... along with a few intermediate values.

"methinks" will never appear. If that were true, any document that happens to contain "methinks" – like this one – could not be English.

This effect comes from Bayes' Theorem, which computes the probability that a document is English, given that you see the word "methinks" to be proportional to the probability of finding "methinks" in an English document. If we estimate the latter as zero, then the former is also zero.

Under that model of English, Shakespeare isn't English. Worse, many documents talking about Shakespeare or that entire era will be excluded from English because of a single mention of "methinks." Even worse, there are thousands of perfectly good English words that do not appear in a typical newspaper corpus, each of which causes the same problem.

Troubles with MLE estimation of underlying frequencies can also appear when you have a language with lots of rare words. There might be 100000 rare words out there, each legal, and each used occasionally, but in a 1000 word corpus, you might catch only 10 of the rare words.

To you, as a user of the corpus, the 10 rare words you catch might seem special, different, more important, but really, they are just the luck of the draw. The words that you catch are are no different from many words you don't catch. In a different corpus, you'd get a different set of ten words. Setting $F_{\text{word}} = 0$ for all the other 99990 rare words is a mistake, because they do occur in English, just not in the sample you collected.

4 Good-Turing estimators

Good-Turing estimators of underlying frequencies are described in other material we have handed out [Gale and Sampson, 1995]. These estimators are based on theory which is correct for a large corpus of a large language. There are many variants on Good-Turing estimators, depending on what way people to choose to solve problems caused by having small corpora or small languages.

All Good-Turing estimators use this equation to calculate the underlying frequencies of events:

$$F_X = \frac{(N_X + 1)}{T} \cdot \frac{E(N_X + 1)}{E(N_X)}, \tag{1}$$

where X is the event, N_X is the number of times you have seen event X, T is the sample size and $E(n)$ is an estimate of how many different events happened exactly n times. Translating that into text-analysis terms, X is a word, N_X is the number of times you have seen word X, T is the size of the corpus and $E(n)$ is an estimate of how many different words were observed exactly n times[3]. Strictly speaking Equation 1 gives the probablity that the *next* word you choose will be X, after looking at a certain corpus.

[3] Following that notation, $E(N_X)$ is an estimate of how many different words were observed the same number of times as word X.

Alternative Notation: Note that many books would write $P(X)$ where we write F_X. Other books would write the same equation as

$$P(X) = r^*/N \tag{2}$$

where

$$r^* = (r+1) \cdot \frac{E(N_{r+1})}{E(N_r)} \tag{3}$$

where (for the two above equations), r is the number of times you've seen word X, N_r is the number of different words that were seen exactly r times, and the $E()$ means you're trying to estimate what N_r would "normally" be, for an infinite corpus of an infinite language. N is the total number of counts, and r^* is called the "adjusted number of observations:" which is how many times you should have seen that word (it is often a fraction).

All the different variants of Good-Turing come from different ways to calculate the $E()$ function. Most importantly, all these variants make sure that $E(N_X)$ is not zero.

So, to pick a concrete example, take a corpus of 30000 English words, our universe is all English words, our event, X, is the word "unusualness." The word "unusualness" appears once, so $N_X = 1$. In a reasonable corpus, you might have 10000 different words that appear once, so $E(1) = 10000$, and you might have 3000 words that appear twice, giving $E(2) = 3000$. The Good-Turing estimate of the probability of "unusualness" is then

$$P(unusualness) = \frac{2}{30000} \cdot \frac{3000}{10000}, \tag{4}$$

which is $2 \cdot 10^{-5}$, in this case, 3 times smaller than the maximum-likelihood value.

Good-Turing estimators give you a total probability for all the data that you have actually observed that is smaller than one. In other words, $\sum_{\text{Observed } X} P(X) < 1$. The remaining bit becomes the probability of seeing something new – some word you haven't seen before. The probability of seeing something new depends on exactly how you calculate E, but is normally about N_1/T.

5 The simplest Good-Turing estimator

The simplest way to do Good-Turing is to pick the function $E()$ so that

$$\frac{E(n+1)}{E(n)} = \frac{n}{(n+1)} \cdot (1 - E(1)/T) \tag{5}$$

Why *Estimates* in Good-Turing?: Why isn't Good-Turing exact? Why are there variants on Good-Turing? The answer is that one expects statistical fluctuations in the number of times that one observes a word.
Now, take some fairly common word that is observed 9,217 times in the corpus, perhaps "Volcker". So, $N_{\text{Volcker}} = 9217$. Let's say that in the Real WSJ Corpus, there are two other words that are observed 9,217 times, so $E^{[observed]}(9217) = 3$.
However, perhaps there are no words that are observed 9,218 times: $E^{[observed]}(9218) = 0$. In that case, we find that Good-Turing (Equation 1) predicts $P(X) = 0$! In other words, that despite the 9217 observations of "Volcker", we are never going to see the word again. A very strange conclusion, but one with a certain logic to it. We have, after all, learned from our corpus that there aren't any words that are observed 9218 times. Why should "Volcker" break that rule?
The reason that "Volcker" (and other words) should break that rule is simply that we know that the rules of English do not constrain how many times a word appears in a document. Still less does English constrain how many times a word appears in a corpus.
Consequently, we need to think of having not just one corpus, but a whole set of equivalent corpora. If the corpus is "all of the Wall Street Journal", then the set of corpora would be derived from versions of the Wall Street Journal in slightly different universes. Mergers and bankruptcies happen in all of them, but different companies, run by different people merge on different days. Some of those corpora will have words that appear 9218 times.
In essence, our estimation rule boils down to imagining what results we might get, if we averaged over a set of "equivalent" corpora. In that thought experiment, we will find that we expect there to be about the same number of words observed 9217 times and 9218 times, because there is nothing magic about the number 9217. $E(9218)$ is then near $E(9217)$. We also expect $E(9218) < E(9217)$ because there are relatively few common words, and many rare words.

This is not the best possible estimate (except for certain special probability distributions), but is good enough for many purposes. Then, we end up with

$$P(X) = \frac{N_X}{T} \cdot (1 - E(1)/T). \tag{6}$$

In other words, we scale down the maximum-likelihood estimator n/T by a factor of $1 - E(1)/T$. That reduces all the probabilities for things we *have* seen, and makes room for things we *haven't* seen.

If we sum over the words we have seen, then the sum of $P(X)$ is (surprise!) $1 - E(1)/T$. Because probabilities sum to one, we have the left-over probability of

$$P(\text{new}) = E(1)/T \tag{7}$$

of seeing something new, where *new* means that you see a new word.

Now, we bring in the universe again. $P(new)$ that we have just calculated is the probability that the next word we pull from the corpus will be *any* English word that we've not seen so far. The probability of catching a *particular* new word will be much smaller.

If we assume that we don't know anything about the English words we haven't yet seen, we can assign them all an equal probability

$$P(X) = \frac{E(1)}{T \cdot U}, \text{ for } N_X = 0, \tag{8}$$

where U is the number of unseen words. Formally,

$$U = E_\Omega - \sum_{i=1}^{\infty} E(i), \tag{9}$$

which says that U (the number of unseen words) is the possible vocabulary (E_Ω, which is the number of distinct words in the language), minus the number of words you've seen once ($E(1)$) minus the number of words you've seen twice ($E(2)$), minus the number of words you've seen three times ($E(3)$), ...

Often, we don't exactly know what E_Ω is, in which case, we either guess, extrapolate in some reasonable way, or try to cast the problem in a form where we can use Equation 7, which doesn't involve U.

6 Expected Likelihood (ELE), Laplace, add-tiny, add-one estimators

All these estimators use the formula

$$P(X) = \frac{N_X + f}{T + E_\Omega \cdot f}, \tag{10}$$

where f is a fudge factor. ELE sets $f = 0.5$, Laplace and add-one estimators are the same thing and set $f = 1$, and add-tiny sets $f = 1/T$.

All of these are useable for Bayes' Theorem purposes. They give the probability $P(X) = f/(T + E_\Omega \cdot f)$ for things you haven't seen.

7 Other Reading

See Mcallester and Schapire [McAllester and Schapire, 2000] for an analysis of the accuracy of Good-Turing estimators.

The first use of Good-Turing is Good [1953], but the origins date back to World War II, in the British project to decode the Enigma cipher: Good [2000].

Also see

- `http://www.grsampson.net/RGoodTur.html`

- http://www.grsampson.net/Resources.html
- http://www.d.umn.edu/~tpederse/Courses/CS8995/Code/sgt-gale.pdf
- http://ei.cs.vt.edu/~history/Good.html
- http://www-rohan.sdsu.edu/~gawron/stat/discounting.htm

Thanks to Peet Morris for comments on this lecture.

References

W. Gale and G. Sampson. Good-turing smoothing without tears, 1995. URL http://citeseer.nj.nec.com/161518.html.

I. J. Good. The population frequencies of species and the estimation of population parameters. *Biometrika*, 40(3/4):237–264, dec 1953. URL http://links.jstor.org/sici?sici=0006-3444%28195312%2940%3A3%2F4%3C237%3ATPFOSA%3E2.0.CO%3B2-K.

I. J. Good. Turing's anticipation of Empirical Bayes in connection with the cryptanalysis of the Naval Enigma. *Journal of Statistical Computation and Simulation*, 66(2), 2000.

David McAllester and Robert E. Schapire. On the convergence rate of good-turing estimators. In *Proceedings of the Thirteenth Annual Conference on Computational Learning Theory (COLT 2000), June 28–July 1, 2000, Palo Alto, California*, 28 June – 1 July 2000. ISBN 1-55860-703-X. URL http://www.learningtheory.org/colt2000/papers/McAllesterSchapire.ps. http://www.learningtheory.org/colt2000/papers/McAllesterSchapire.ps .