Maximum Entropy Good-Turing Estimator for Language Modeling

Juan P. Piantanida, Claudio F. Estienne

School of Engineering
University of Buenos Aires, Argentina
jpianta,cestien@fi.uba.ar

Abstract

In this paper, we propose a new formulation of the classical Good-Turing estimator for n-gram language model. The new approach is based on defining a dynamic model for language production. Instead of assuming a fixed probability distribution of occurrence of an n-gram on the whole text, we propose a maximum entropy approximation of a time varying distribution. This approximation led us to a new distribution, which in turn is used to calculate expectations of the Good-Turing estimator. This defines a new estimator that we call Maximum Entropy Good-Turing estimator. Contrary to the classical Good-Turing estimator it needs neither expectations approximations nor windowing or other smoothing techniques. It also contains the well know discounting estimators as special cases. Performance is evaluated both in terms of perplexity and word error rate in an N-best re-scoring task. Also comparison to other classical estimators is performed. In all cases our approach performs significantly better than classical estimators.

1. Introduction

It is a well known fact that state-of-the-art speech recognition systems uses n-gram models in their language models. In order to estimate such models, it is necessary to use probability estimators which assign a probability to each n-gram. Because of the sparse characteristic of language two problems often arise. On the one hand the number of samples of a particular event is often inadequate to obtain robust estimators of such event. On the other hand, even when the amount of available training data is huge, many events do not occur at all, but this does not mean they have zero probability of occurrence, it just means they did not occur in the training set. As a consequence the maximum likelihood estimator of the probability given by the quotient r/n where r is the frequency of occurrence of an event (n-gram) and N is the total number of events, will not be in general a good estimator of the probability. On one hand it will assign null probability to non zero occurrence events, on the other hand it can be shown [10] that it tends to over-estimate events which have low frequency of occurrence in a text. In order to deal with the problem of sparseness of data, many probabilities estimators have been proposed on the literature. Two of the most popular are the Good-Turing estimator [3], [6] and discounting estimators [2], [1].

In this work we take a different approach. We assume a dynamic language model for speech production in the sense that the frequency of occurrence of an event is not fixed on the text, but is a random variable. Even when this view requires a careful mathematical treatment, it is possible using maximum-entropy models to obtain an approximation which requires an estimator which just depends on r. Starting with classical Good-Turing estimator, we will re-formulate it in order to meet our model requirements. As a result a new estimator called maximum entropy Good-Turing estimator will be obtained. This new estimator does not need approximations and empirical formulations as in the case of classical Good-Turing estimator [3], [7].

In the next section we briefly describe classical Good-Turing estimation and maximum-entropy models in order to understand our formulation. In section 3 we formally state our Good-Turing maximum entropy model and we discuss some issues related to it. Experimental results are shown in section 4. Finally some concluding remarks are given in section 5.

2. Classical Good-Turing estimator and maximum entropy models

2.1. Good-Turing estimator

Classical Good-Turing estimator [3] can be stated as a formal model [7], [6] in which the probability of an event σ (an n-gram) whose frequency of occurrence r is given by: $P(\sigma) = q_r$, with:

$$q_r = \frac{r^*}{N} \tag{1}$$

where:

$$r^* = (r+1)\frac{\mathcal{E}_{r+1,c_{r+1}}\{c_{r+1}\}}{\mathcal{E}_{r,c_r}\{c_r\}}$$

$$\mathcal{E}_{r+1,c_{r+1}}\{c_{r+1}\} = \sum_{\forall c_{r+1}} c_{r+1} P(r+1, c_{r+1}) \tag{2}$$

$$\mathcal{E}_{r,c_r}\{c_r\} = \sum_{\forall c_r} c_r P(r, c_r) \tag{3}$$

r is the frequency of repetition of an event, N is the total number of events, C_r corresponds to the number of events whose frequency of occurrence is r, and $P(r, c_r)$ is the joint probability distribution of C_r events with frequency r. A fundamental hypothesis of the model is the symmetry requirement which states that any two events having the same frequency in the text must also have the same probability estimate [6]. Equations (2) and (3) are difficult to determinate and they are not used in practical implementations of the Good-Turing estimator, instead they are approximated with training data. As a consequence, many values of c_r are zero, and there exists an unacceptable dispersion between values of c_r and c_{r+1}. These problems make necessary the use of windowing techniques, or non continuous q_r in order to smooth such dispersions [7]. Even though smoothing is necessary, in practical implementations, not only mathematical formality is lost with this approximation, but also empirical adjustments are necessary for each kind of text.

2.2. Maximum entropy models

Maximum-entropy models have been used in language model contexts to estimate n-grams (see for example [11]), basically they can be stated as follows:

- Reformulate the different information sources as constrained to be satisfied by the target estimate.
- Among all probability distributions that satisfy these constraints, choose the one that has the highest entropy.

Mathematically m constrains are expressed as expectation functions as follows:

$$\mathcal{E}\{g_k(x)\} = \sum_{\forall x_i} g_k(x_i)P(x_i) \quad k = \{1, \ldots, m\} \tag{4}$$

$g_k(x)$ are model constrains usually expressed as expectation of these functions. The distribution which maximize entropy given such constrains is given by [12]:

$$p(x) = \frac{\exp\left(-\sum_{k=1}^{m} \lambda_k g_k(x)\right)}{\mathcal{Z}(\lambda_1, \ldots, \lambda_m)} \tag{5}$$

where $\mathcal{Z}$ is the partition function.

3. Maximum entropy Good-Turing estimator

3.1. A dynamic model for language production

We can think of the speech production process as follows, consider a hypothetical speaker who starts to speak to another person about some specific topic, at this moment his vocabulary is reduced to the number of words he said up to a particular moment t_1 say N_{t_1}. the number of repetitions is expected to be low at first so, a reasonable assumption for the probability of emission of a word is $1/N_{t_1}$. If we use entropy as a measure of the information of the message at time t_1, it will be approximately $H_{t_1} \cong \log N_{t_1}$ [12]. After some time of emitting words, say at instant t_2, speaker vocabulary will increase to N_{t_2} and, language entropy will also grow. However at this point, some vocabulary repetitions are expected to have occurred, decreasing the grown rate of entropy. As a consequence, H_{t_2} will be lower than $\log N_{t_2}$. Our assumption is that in the long term, language entropy of that dynamic process, will grow at decreasing rate up to a maximum stationary value. This value would correspond to the case when the speaker has used nearly all his vocabulary concerning a specific topic to a specific person, and the number of repetitions is enough to avoid growth entropy any more.

It means that we are viewing language production as a dynamic process by which the probability of an event is not fixed but is a function of time, so it could be zero at a moment (when no examples of an event are emitted up to that moment), and non-zero at another moment. A complete formulation of the dynamics of this model is out of the scope of the present work however, if we assume that in the long term the system bounds a maximum entropy state which does not change any more, a simplified model can be developed and a robust estimator of the probability of an event found.

3.2. Model constrains

It should be clear from the discussion above, that r, the frequency of occurrence of event, is not constant but it changes when speaker introduces more and more vocabulary. We can think of it as a random variable with an associated probability $P_t(r)$ which, of course, is unknown. Index t means distribution changes with time. If we adopt the symmetry requirement used in Good-Turing estimator, we will not be able to distinguish between different events that occur the same number of times, so the distribution which represents model dynamics will not only be a function of r, but also the number of events whose frequency of occurrence is r. If we call such number c_r, we will have an associated distribution $P_t(r, c_r)$. But we are not interested in the instantaneous dynamics of the model, instead we are concerned with the distribution whose entropy reaches a stable maximum. Such distribution would corresponds to the best static approach we could do of our dynamic process. We will call such distribution $P(r, c_r)$.

In order to find $P(r, c_r)$ we will embody four statistics that include information of the process necessary for the model. The first is:

$$\mathcal{S}_1 = \sum_{\forall\, \sigma} N(\sigma) \tag{6}$$

where σ is an event, and $N(\sigma)$ is the number of times such event occurs. This statistics corresponds to a sufficient statistics for the Poisson distribution [12]. The choice of this statistics is based on a previous work [9] which shows that the frequency of occurrence of an event in a text follows a Poisson distribution. In another work [8], it is also shown that c_r, (the number of events with frequency r), also responds to a Poisson distribution, but different for each r, so the second statistics that we incorporate is:

$$\mathcal{S}_2 = \sum_{k=0}^{N_r} \sum_{\forall\, \sigma} \delta(N(\sigma), k) \tag{7}$$

where N_r is the maximum number of occurrences of an event and $\delta(i, j) = 0 \;\; \forall i \neq j$. We also define two statistics which take into account dynamics properties:

$$\mathcal{S}_3 = \sum_{k=0}^{N_r} \sum_{\forall\, \sigma} k\delta(N(\sigma), k) \tag{8}$$

$$\mathcal{S}_4 = \sum_{\forall\, \sigma} \log N(\sigma) \tag{9}$$

Now we are ready to formulate a maximum entropy probability distribution $P(r, c_r)$ that meets our four constrains.

3.3. Calculus of the distribution

Our four statistics (6), (7), (8) and (9) are put together in the model trough equation (4) resulting:

$$\sum_{r=1}^{N_r} \sum_{c_r=0}^{N_c} rP(r, c_r) = \langle r \rangle \tag{10}$$

$$\sum_{r=1}^{N_r} \sum_{c_r=0}^{N_c} c_r P(r, c_r) = \langle c_r \rangle \tag{11}$$

$$\sum_{r=1}^{N_r}\sum_{c_r=0}^{N_c} rc_r P(r,c_r) = \langle rc_r \rangle \tag{12}$$

$$\sum_{r=1}^{N_r}\sum_{c_r=0}^{N_c} \log(r) P(r,c_r) = \langle \log r \rangle \tag{13}$$

Where $\langle \log r \rangle$, $\langle c_r \rangle$, $\langle rc_r \rangle$ y $\langle r \rangle$ are evaluated from training data, N_r is the maximum number of occurrences for all event and N_c is the maximum number of events that occur r times with the same frequency. Maximizing the entropy of $P(r,c_r)$ with the above constrains we obtain the corresponding equation (5) related to our model:

$$P(r,c_r) = \frac{r^{-\lambda_1} e^{-c_r(\lambda_2+\lambda_3 r)} e^{-\lambda_4 r}}{\mathcal{Z}(\lambda_1,\lambda_2,\lambda_3,\lambda_4)} \tag{14}$$

where:

$$\mathcal{Z}(\lambda_1,\lambda_2,\lambda_3,\lambda_4) = \sum_{r=1}^{N_r}\sum_{c_r=0}^{N_c} r^{-\lambda_1} e^{-c_r(\lambda_2+\lambda_3 r)} e^{-\lambda_4 r}$$

As said, expectations $\langle \log r \rangle$, $\langle c_r \rangle$, $\langle rc_r \rangle$ y $\langle r \rangle$, are obtained from training data. We have used re-sampling statistical techniques which give rise to Jackknife's estimators [13] however, other techniques could have been used. Once we get expectations we can get parameters λ_1, λ_2, λ_3 y λ_4 using IIS algorithm [5]. Finally applying formula 14 we get our maximum entropy distribution. The next step is to introduce this distribution in the Good-Turing estimator.

3.4. Maximum entropy Good-Turing estimator

Once (14) is determined, it is not difficult to calculate expectations of the Good-Turing estimator (1). It is straightforward to show that:

$$\mathcal{E}_{r,c_r}\{C_r\} = \frac{K r^{-\lambda_1} e^{-(\lambda_2+\lambda_3 r)} e^{-\lambda_4 r}}{\left(1-e^{-(\lambda_2+\lambda_3 r)}\right)^2} \tag{15}$$

Finally replacing (15) in (1) we obtain our new maximum entropy Good-Turing estimator:

$$q_r = \frac{(r+1)}{N}\left(\frac{r}{r+1}\right)^{\lambda_1} \left(\frac{1-e^{-(\lambda_2+\lambda_3 r)}}{1-e^{-(\lambda_2+\lambda_3(r+1))}}\right)^2 e^{-(2\lambda_3+\lambda_4)} \tag{16}$$

3.5. discussion

It is important to compare our estimator with maximum-likelihood estimator $q_r = r^*/N$, defining the quotient r^*/r:

$$\frac{r^*}{r} = \left(\frac{r+1}{r}\right)\left(\frac{r}{r+1}\right)^{\lambda_1} \left(\frac{1-e^{-(\lambda_2+\lambda_3 r)}}{1-e^{-(\lambda_2+\lambda_3(r+1))}}\right)^2 e^{-(2\lambda_3+\lambda_4)}$$

This quotient allows us to understand the influence of the parameters model. Parameter λ_1 is a measure of the velocity of growing of $P(r,c_r)$ when r increases. Parameter λ_2 is related to the value of the estimator at very low values of r (including $r=1$). Parameter λ_3 measures the maximum likelihood limit our estimator will reach. Finally, parameter λ_4 is related to a multiplicative factor (independent of r). This parameter will affect the probability mass of unobserved events. If we think unobserved events probability as:

$$P(\varphi_0) = q_0 C_0 = 1 - \sum_{r=1}^{N_r} q_r c_r$$

an increase of the parameter λ_4 will decrease q_r, as a consequence $P(\varphi_0)$, the probability of unobserved events will also grow.

Another advantage of our estimator is that it verifies two desired requirements for an estimator [1]: $q_r \leq r/N$, and $q_{r-1} \leq q_r \ \forall r$. the second condition is easily seen from (16), in order to verify the first condition we have found an equivalent condition to $q_r \leq r/N$ which is verified by our estimator:

$$\left(\frac{r}{r+1}\right)^{\lambda_1 + 2\lambda_3\lambda_2 e^{-\lambda_2} - 1} \left(\frac{1}{2\lambda_3}\right)^{e^{-\lambda_4}} < 1$$

Finally, if we make a series expansion of expression (16) and we take the linear term, also making a convenient choice of parameters λ_1, λ_2,λ_3 and λ_4, Ney discounting estimators [1] results as a special case of the maximum entropy Good-Turing estimator

4. Experimental results

4.1. data description

Experiments were performed on three corpora: an English database, switchboard phase one, and two Spanish databases, Latino 40 (available from LDC) and Latin-American Spanish database collected by SRI International [4]. We also used text extracted from newspapers. We performed perplexity measurements using the whole databases, and N-best re-scoring using switchboard corpus. We used bi-gram models with Latino40 corpus and tri-gram models with switchboard and Latin-American Spanish databases. The whole text was split in three classes:

- Text A: It consists of text taken from Latino40 transcriptions, we used 32k words for training and 8k words for testing.
- Text B: It consists of text taken from Latin-American Spanish database transcriptions and newspapers texts, combining both classes of text we used 752k words for training, and 33k words for testing.
- Text C: It consists of 3M words taken from switchboard phase one transcriptions used for training, and 59k words taken from hub-5 2001 evaluation set transcriptions used for testing.

4.2. results

Perplexities measurements were performed over classical Good-Turing estimator (CGT) [3], Katz estimator (KATZ) [2], Absolute discounting (ADE) and linear discounting (LDE) estimators [1] and Maximum entropy Good-Turing (MEGT). Results can be shown in table 1.

Estimator	Text A (bigram)	Text B (trigram)	Text C (trigram)
CGT	219	739	534
ADE	149	251	160
LDE	138	693	176
KATZ	156	232	155
MEGT	**134**	**218**	**146**

Table 1: *Perplexities of different estimators with different vocabulary*

Finally we performed N-best re-scoring over 5895 sentences corresponding to HUB-5 2001 test set. We re-scored 2000-best hypothesis performed by The SRI DECIPHER(TM) speaker-independent continuous speech recognition system at SRI International. Results are shown in table 2

estimator	wer
BASELINE	31.8
KATZ	31.5 (0.9%)
MEGT	**30.7 (3.4%)**

Table 2: *WER after re-scoring using Katz and MEGT estimators.*

4.3. discussion

Table 1 shows maximum entropy method reports an improvement in terms of perplexity superior to the rest of the estimators. It is interesting to observe that, improvement is performed over all three text corpora. This is an important difference in respect off the other estimators. For example Katz estimator has lower perplexity for text B and text C than for text A.

Table 2 shows results on N-best re scoring over switchboard corpus in terms of WER. Only Katz estimator gave a small improvement, the rest of estimators were not included because they did not decrease baseline WER. We can see a significant improvement concerning the baseline of 3.4% in the our maximum entropy Good-Turing estimator. We could expect a greater increase if we use maximum entropy estimator in a n-gram model on a ASR task.

5. Conclusions

Using maximum entropy method and assuming a dynamic model for language production, we have found a Good-Turing like estimator which does requires neither smoothing nor empirical adjustments which are necessary in the classical Good-Turing estimator. Parameters defining our model are determined using the well known IIS algorithm. We also have shown our new estimator verify both requirements desired in language estimators:$q_r \leq r/N$, and $q_{r-1} \leq q_r \ \forall r$. Finally we shown that our estimator contains Ney discounting estimator as a particular case.

Experimental results show maximum entropy method performs better than all others estimators for the three classes of text corpora considered. We also tested our estimator in a 2000 hypothesis N-best re scoring over switchboard corpus obtaining decrements in the WER of 3.4% refered to the baseline.

6. Acknowledgments

We want to thank Star-Lab at SRI International and specially Dr. Horacio Franco for permitting the use of their Latin-American Spanish database, and N-best data. We also thank Luciana Ferrer from SRI for her comments and suggestions.

7. References

[1] Ney, H., Essen, U., and Kneser, R., “On the Estimation of ’Small’ Probabilities by Leaving-One-Out” IEEE Trans. on Pattern Analysis and Machine Intelligence, 17(12): 1202–1212, 1995.

[2] Katz, S. M., “Estimation of probabilities from sparse data for language model component of a speech recognizer”, IEEE Trans. on Acoustics, Speech and Signal Proc., 35(3):400–401, 1987.

[3] Good, I. J., “The population frequencies of species and the estimation of population parameters”, Biometrika, Vol. 40:237–264, 1953.

[4] Bratt, H., Neumeyer, L., Shriberg, E., Franco, H., “Collection and Detailed Transcription of a Speech Database for Development of Language Learning Technologies” Proc.. ICSLP, Sydney, Australia, December 1998.

[5] Della Pietra, S., Della Pietra, V., and Lafferty, J., “Inducing Features of Random Fields” IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. PAMI-19, p.380, 1997.

[6] Nadas, A., “On Turing’s formula for word probabilities” IEEE Trans. on Acoustic, Speech and Signal Proc., 33(12):1414–1416, 1985.

[7] Gale, W., “Good-Turing Smoothing Without Tears” Report AT&T Bell Laboratories, 2000.

[8] Witten, I. H., and Bell, T. C., “The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression” IEEE Trans. on Information Theory, 37(12):1085–1094, 1991.

[9] Church, W. K., and Gale, W. A., “Poisson mixtures” AT&T Bell Labs-Research.

[10] Lindsey, J. K., and Denne, J. S., “Missing data: a fundamental frequentist problem” Report Biostatistics, Limburgs University, Diepenbeek, Belgium

[11] Rosenfeld, R., “A Maximum Entropy Approach to Adaptive Statistical Language Modeling” Computer Speech and Language, 10(3)187–228, 1996.

[12] Cover, T. and Thomas, J., “Elements of Information Theory” John Wiley & Sons, New York, NY, 1991.

[13] Walsh, B., “Re sampling methods: randomization test, Jackknife and Bootstrap Estimators” Lecture Notes, 2000.