Estimating Probabilities from Small Samples

Bruno Jedynak, Sanjeev Khudanpur and Ali Yazgan
Johns Hopkins University, Baltimore, MD 21218, USA,
Emails: {bruno.jedynak,khudanpur,ayazgan1}@jhu.edu

Abstract:

A novel solution is presented to a recurring problem in statistical modeling—estimating a probability mass function (pmf) for a discrete random variable from a small sample. The solution naturally leads to smooth pmf estimates, requires no held out data, nor makes any prior assumptions about the unknown pmf, while still providing a way to incorporate prior knowledge when available. A pmf is deemed *admissible* as an estimate if it assigns merely a higher likelihood to the observed value of a sufficient statistic than to any other value possible for the same sample size. The maximum likelihood estimate is trivially admissible by this definition, but so are many other pmfs. An estimate is selected from this admissible family via criteria such as maximum entropy or minimum I-divergence. Empirical results in statistical language modeling are presented to demonstrate that estimates obtained in this manner have performance that is competitive with state-of-the-art estimates, and have additional desirable properties not found in the state-of-the-art.

1. Common Responses to Data Sparseness

Let P_{True} be a probability mass function (pmf) on a discrete and finite set $\mathcal{V} = \{1, \ldots, V\}$, and consider the problem of estimating P_{True} from a sequence $\mathcal{W} = \{w_1, \ldots, w_N\}$ of independent samples, drawn according to a common distribution P_{True}. To facilitate a more concrete exposition, think of $\mathcal{V}$ as the vocabulary of a statistical language model (LM), and $\mathcal{W}$ as the training corpus. This estimation problem is, of course, a recurring problem not only in natural language processing (NLP) but indeed in all of statistics. A popular estimate of P_{True} is the maximum likelihood estimate,

$$\hat{P} \equiv \hat{P}(v) = \frac{c_v}{N} = \frac{1}{N}\sum_{t=1}^{N} \mathbf{I}(w_t = v), \quad \forall v \in \mathcal{V}, \quad (1)$$

where $\mathbf{I}(A)$ is the indicator function of an event A, and c_v is the observed count of a word v in the corpus $\mathcal{W}$. $\hat{P}$ is usually adequate when $N \gg V$ and $N \gg \max_v \left\{\frac{1}{P_{\text{True}}(v)}\right\}$. But small samples usually result in $\hat{P}(v)$ being unacceptably small for some $v \in \mathcal{V}$. Zero probabilities, in particular, lead to severe performance degradation when the estimate $\hat{P}$ is subsequently employed, e.g., for automatic speech recognition, parsing, machine translation, and other NLP applications in which statistical models are used.

One standard solution to data sparseness is Bayesian estimation, in which P_{True} itself is viewed as a continuous-valued random variable on the unit simplex $\mathbb{P} \subset \mathbb{R}^V$, with a prior probability density $\pi(P)$. Under a quadratic loss function, the Bayesian estimate $\hat{P}_{\text{Bayes}}$ of P_{True} is the mean of the posterior probability $\pi(P|\mathcal{W})$ given the sample. Formulae such as

$$\hat{P}_{\text{Bayes}} \equiv \hat{P}_{\text{Bayes}}(v) = \frac{c_v + \alpha}{N + \alpha V}, \quad \forall v \in \mathcal{V}, \quad (2)$$

for the Dirichlet prior have been used, where the *hyperparameter* α is chosen based on some prior knowledge about P_{True}; $\alpha = 1$, $\alpha = \frac{1}{2}$ and $\alpha = \frac{1}{V}$ are often used [1]. The choice of the prior itself, such as the Dirichlet, is sometimes guided by the tractability of the resulting computation of the mean under the posterior probability.

Other well known solutions in language modeling (and elsewhere) include Good-Turing discounting, constant discounting, natural discounting, Witten-Bell discounting, and others [2]. In all these methods, the discounted estimate of P_{True} is computed as

$$\hat{P}_{\text{Discount}} \equiv \hat{P}_{\text{Discount}}(v) = \begin{cases} \frac{c_v - \delta(c_v)}{N} & \text{if } c_v > 0, \text{ and} \\ \frac{\delta(0)}{N} & \text{if } c_v = 0, \end{cases} \quad (3)$$

where the *discount* parameters $\delta(\cdot)$ are estimated via some heuristics, or from a held out portion of $\mathcal{W}$, to provide nonzero probability to unseen words. Readers interested in details of these methods are referred to the survey paper by Chen and Goodman [3]. All these methods make some ad hoc assumptions about the unknown P_{True} that are not substantiated in $\mathcal{W}$, and some methods additionally divide $\mathcal{W}$ into a training and a held out set, further aggravating data sparseness. Even in the theoretically pleasing Bayesian case, the purpose of the prior is often only to ensure that the MAP estimate $\hat{P}_{\text{Bayes}}$ is positive everywhere, and different priors that each ensure this may therefore lead to different estimates.

Maximum entropy estimation is another standard solution to data sparseness. Instead of estimating $\hat{P}(v)$ for each $v \in \mathcal{V}$ according to (1), the maximum entropy method first estimates $\hat{P}(v \in A_j) = \hat{a}_j$ for *select sets* $A_j \subset \mathcal{V}$, $j = 1, \ldots, J$, for which we have sufficient evidence in $\mathcal{W}$. Fixing the probability of some subsets

of $\mathcal{V}$ in this manner typically under-specifies the pmf of interest, leading to a set $\mathcal{P}$ of *admissible* pmfs

$$\mathcal{P} = \{P \in \mathbb{P} : P(A_j) = \hat{a}_j,\ j = 1, \ldots, J\} \quad (4)$$

in which the maximum likelihood estimate $\hat{P}$ of (1) is but one member. The admissible pmf with the highest Shannon entropy is then chosen as the estimate of P_{True}:

$$\hat{P}_{\text{MaxEnt}} = \arg\max_{P \in \mathcal{P}} \sum_{v \in \mathcal{V}} P(v) \log \frac{1}{P(v)}. \quad (5)$$

It can be shown that for every $v \in \mathcal{V}$, as long as at least one $P \in \mathcal{P}$ satisfies $P(v) > 0$, it follows that $\hat{P}_{\text{MaxEnt}}(v) > 0$. Thus the maximum entropy estimate is inherently smooth. There are several heuristics but few principles for selecting the sets $A_j \subset \mathcal{V}$ or even J. Some A_j's are typically singleton, specifying, for instance, the probability of words that have been seen sufficiently often in $\mathcal{W}$; some A_j's may contain all words which can take on a certain grammatical part of speech, e.g. adjective; some A_j's may overlap with others; etc. Therefore, while maximum entropy estimation (5) eliminates the need for some of the ad hoc assumptions made by other techniques, it leaves open the problem of selecting the (feature) sets that define $\mathcal{P}$ in (4).

Another weakness of the classical maximum entropy method is that the specification of $\mathcal{P}$ via equality constraints leads to an *ad hoc* choice for any candidate A_j — either one must constrain its probability to be *exactly* $\hat{a}_j$, or leave it completely unconstrained. This is unsatisfactory. For instance, if one were considering all singleton sets $\{v\}$ as candidate A_j's, then naively including all of them in the definition of $\mathcal{P}$ amounts to requiring $P(v) = \hat{P}(v)$ for all v, or $\mathcal{P} = \{\hat{P}\}$. On the other hand, not imposing constraints on, say, all sets $\{v\}$ with $c_v \leq 1$ may result in $\hat{P}_{\text{MaxEnt}}(u) = \hat{P}_{\text{MaxEnt}}(v)$ even if $c_u = 0$ and $c_v > 0$. Inequality constraints have been proposed to address this problem [4, 5].

$$\mathcal{P} = \{P : a_j \leq P(A_j) \leq b_j,\ j = 1, \ldots, J\}. \quad (6)$$

To the best of our knowledge, there has not been much discussion in the literature of a principled way to choose a_j and b_j, particularly in a way that depends only on $\mathcal{W}$.

Some techniques take inspiration from the exponential form of the maximizer $\hat{P}_{\text{MaxEnt}}$ in (5), and attempt to address the $\mathcal{P} = \{\hat{P}\}$ problem by assuming a prior on the parameters of the exponential form, a prior that excludes $\hat{P}$ as a solution [6]. They are Bayesian estimates (2) of a parametric pmf, not maximum entropy estimates (5) of P_{True}, and entail *ad hoc* hyperparameters just as other Bayesian methods.

In the following section, we briefly describe a technique we have recently developed to address some of these concerns using a notion of a *maximum likelihood set* [7]. In Section 3., we describe how this technique is applied to the estimation of a statistical language model. We present empirical results in Section 4. and conclude in Section 5..

2. The Maximum Likelihood Set

Let $\hat{\mathbb{P}}^{(N)} \subset \mathbb{P}$ denote the set of all possible empirical distributions, or *types*, for a sample of size N [8]. A type, which is the same as the maximum likelihood estimate of (1), is fully specified by the counts $(c_1, \ldots, c_V)$, and for N independent samples drawn according to a common pmf $P_{\text{True}} \in \mathbb{P}$, the probability of observing a type $\hat{P} \in \hat{\mathbb{P}}^{(N)}$ is

$$\begin{aligned} P_{\text{True}}(\hat{P}) &= P_{\text{True}}(c_1, \ldots, c_V) \\ &= \frac{N!}{c_1! \ldots c_V!} \prod_v P_{\text{True}}(v)^{c_v} \end{aligned} \quad (7)$$

For a given type $\hat{P} \in \hat{\mathbb{P}}^{(N)}$, we define the *maximum likelihood set* (MLS) as

$$\mathcal{M}(\hat{P}) = \left\{ P \in \mathbb{P} \,|\, P(\hat{P}) \geq P(\hat{P}'), \forall \hat{P}' \in \hat{\mathbb{P}}^{(N)} \right\} \quad (8)$$

In words, $\mathcal{M}(\hat{P})$ is the set of all pmfs under which the observed type is no less likely than any other type in $\hat{\mathbb{P}}^{(N)}$. An equivalent characterization of the MLS is provided in [7].

$$\begin{aligned} \mathcal{M}(\hat{P}) = \{P \in \mathbb{P} : \forall (u, v) \in \mathcal{V} \times \mathcal{V}, \\ (c_u + 1)P(v) \geq c_v P(u)\} \end{aligned} \quad (9)$$

As $\hat{P}$ ranges over types in $\hat{\mathbb{P}}^{(N)}$, the MLSs cover all of $\mathbb{P}$, as illustrated for $V = 3$ in Figure 1. The MLS has several desirable properties which we reproduce from [7].

1. $\mathcal{M}(\hat{P})$ is a closed, convex, $V \times (V-1)$ sided polyhedron in $\mathbb{P}$.
2. The observed type $\hat{P}$ is always in $\mathcal{M}(\hat{P})$ but no other type in $\hat{\mathbb{P}}^{(N)}$ is.
3. Diameter: $\|P - \hat{P}\|_1 \leq 2(V-1)N^{-1}$ for every $P \in \mathcal{M}(\hat{P})$.
4. Viewing $\hat{P}$ as a random variable, $\lim_{N \to 0} \sup_{P \in \mathcal{M}(\hat{P})} \|P - P_{\text{True}}\|_1 = 0$, P_{True}-a.s.
5. If $c_v > 0$ for some word v then $P(v) > 0$ for every $P \in \mathcal{M}(\hat{P})$.

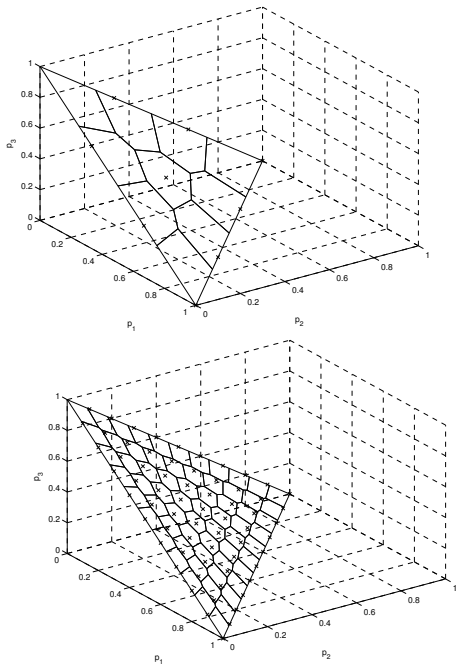

Figure 1: Illustration of the Maximum Likelihood Sets in $\mathbb{P}$ for all the possible types for an alphabet of size $V = 3$ for $N = 3$ samples (Left) and $N = 10$ samples (Right). Each polygonal "cell" is an MLS contains exactly one type marked with a cross.

6. There exist pmfs $P \in \mathcal{M}(\hat{P})$ such that $P(v) > 0$ for every $v \in \mathcal{V}$.

7. Faithfulness to evidence: if $c_u < c_v$ then $P(u) \leq P(v)$ for every $P \in \mathcal{M}(\hat{P})$.

$\mathcal{M}(\hat{P})$ is proposed as an *admissible* set of pmfs, from which a particular pmf may be chosen as an estimate of the underlying pmf P_{True} using secondary criteria. In particular, if a *reference pmf* Q is available, i.e. an estimate one would find acceptable when $N = 0$, then one way to choose an element of $\mathcal{M}(\hat{P})$ is :

$$\begin{aligned} \hat{P}^*_\mu &= \arg \min_{P \in \mathcal{M}(\hat{P}_{ML})} D(P\|Q) \\ &= \arg \min_{P \in \mathcal{M}(\hat{P}_{ML})} \sum_{v \in \mathcal{V}} P(v) \log \frac{P(v)}{Q(v)} \end{aligned} \quad (10)$$

Attainment of the minimum in (10), and the uniqueness of the minimizer, is guaranteed by $\mathcal{M}(\hat{P})$ being closed and convex and by the convexity of $D(p\|q)$ [9, Theorem 2.1].

Now, if Q is the uniform pmf on $\mathcal{V}$, then the criterion (10) for selecting $\hat{P}^*_\mu$ is simply maximum entropy. Note further that the K-L divergence is well defined, and retains its convexity, even if Q does not sum to unity. This enables encoding prior knowledge about the relative probabilities of words in Q with considerable ease. Finally, $c_u = c_v$ and $Q(u) = Q(v)$ guarantees $\hat{P}^*_\mu(u) = \hat{P}^*_\mu(v)$. This results in great simplifications in the numerical computation of $\hat{P}^*_\mu$ in our experiments.

3. Applications to Statistical Language Modeling

Statistical language models are a key component in NLP applications such as automatic speech recognition, machine translation, spelling correction, and document retrieval. Language modeling entails estimating a probability distribution over word-sequences, and this is typically done by modeling the sequence of words in a sentence by a finite memory Markov chain. An n-gram model is a set of conditional pmfs $P(w_n|w_1, \ldots, w_{n-1})$, one for every conditioning event. In applications such as document retrieval, where word-order is not of paramount importance and a bag-of-words representation is adequate, i.i.d. models, called unigram models, are used. In all cases, there is a need to estimate a pmf, marginal or conditional, on the vocabulary.

In this section, we present experimental results for the estimation of unigram models as well as joint probabilities for pairs of consecutive words, or bigrams. We begin, for the sake of completeness, with a brief review of standard unigram and bigram estimation using a smoothing technique popular both in language modeling and elsewhere in statistics.

3.1 Good-Turing Estimation of Probability Mass Functions

Given a training sample $\mathcal{W}$ for the estimation of a unigram model, let n_c denote the number of words with count value c. Note that $0 \leq n_c \leq V$ for $0 \leq c \leq N$ and n_c is zero otherwise. The Good-Turing estimate of P_{True} [10], which may be heuristically justified in many ways, is given by

$$\hat{P}_{\text{GT}}(v) = \frac{(c_v + 1)n_{c_v+1}}{N n_{c_v}}, \qquad \forall v \in \mathcal{V}. \quad (11)$$

The Good-Turing formula may be written as

$$\hat{P}_{\text{GT}}(v) = \frac{(c_v + 1)n_{c_v+1}}{c_v n_{c_v}} \hat{P}(v), \qquad \forall v : c_v > 0, \quad (12)$$

and since the term in front of $\hat{P}(v)$ is typically less than 1, particularly for small c_v, the technique is called Good-Turing "discounting." We will compare the empirical performance of (11) with the corresponding estimate of (10) for several different choices of Q.

Note that all words which have the same count in the training corpus get the same probability in (11) and (12).

In particular, the Good-Turing estimate (11) resembles the maximum likelihood estimate (1), except that it uses smoothed counts

$$c_v^* = \frac{(c_v + 1)n_{c_v+1}}{n_{c_v}} \tag{13}$$

instead of the original counts c_v. This view permits extension of the Good-Turing formula to joint probabilities of word-pairs, word-triples etc. E.g., if c_{uv} denotes the number of times u was followed by v in a training corpus, then Good-Turing bigram estimate is

$$\hat{P}_{\rm GT}(u,v) = \frac{(c_{u,v}+1)n_{c_{u,v}+1}}{Nn_{c_{u,v}}}, \qquad \forall u,v \in \mathcal{V}\times\mathcal{V}. \tag{14}$$

Other methods for smoothing bigram distributions are discussed at length elsewhere [3].

3.2 Reference Distributions for MLS-Based Estimation of Word Probabilities

If obtaining smooth estimates was the primary goal, one would naturally use the uniform distribution on $\mathcal{V}$ for Q in (10). We obtain empirical results for this (maximum entropy estimation) case as a first step. However, all words are not equally likely *a priori*, and it is known from several studies that the count c_v and the *rank* of a word v, when the vocabulary is sorted in order of decreasing counts, has a roughly inverse relationship named Zipf's law [11]. This provides a natural *reference pmf* Q for (10). Specifically, empirical studies suggest that

$$Q(v) = \frac{1}{\text{rank}^\beta(v)} \tag{15}$$

holds with $\beta = 1$ for unigrams and $\beta = 0.65$ for bigrams [12]. Note that the normalization constant that makes Q a pmf need not be computed, since it plays no role in the minimization of (10). The resulting estimate $\hat{P}_\mu^*$ may then be interpreted as *the pmf closest to Zipf's law that is supported by the evidence in* $\mathcal{W}$.

Given a vocabulary, however, we still have no *a priori* way of determining the rank-ordering of words needed for (15). We simply use the rank-ordering empirically observed in $\mathcal{W}$ to determine Q. Furthermore, instead of breaking ties between words with the same count in $\mathcal{W}$, all words which have the same count get a rank equal to the mean of the ranks spanned by them. This last simplification results in a significant reduction in complexity. By assuming words with the same observed counts to have the same Q-probability, we are assured that they will have the same $\hat{P}_\mu^*$ probability, reducing the number of free variables in the numerical optimization of (10) of finds $\hat{P}_\mu^*$. In the unigram pmfs estimated below, for instance, the number of parameters needed to specify $\hat{P}_\mu^*$ reduces from 50,000 to 600. The minimizer of (10) is computed using the numerical optimization package CFSQP [13].

4. Empirical Results for Language Modeling

We have conducted experiments on English text from the Wall Street Journal corpus. A particular subset of this corpus, called the UPenn Treebank corpus, widely used by many researchers in language modeling, has a standard division into Sections, named 00 through 24. We use Sections 00-22, containing about 900K words, as our training corpus, and Sections 23-24, containing 100K words comprise our test corpus. For the purpose of studying the variability of the estimates, we divide Sections 00-22 into 10 roughly equal parts, and results averaged over these smaller training corpora are also presented.

We make a list of all seen words from Sections 00-22 and augment this vocabulary with a set of unspecified "unseen" words. The decision on how many unseen words to include is presently ad-hoc — based on a leave-one-out estimate, the number of unseen words is set exactly equal to the number of words seen only once in the corpus. Every new word in the test set, upon first encounter, *displaces* one of the hitherto unspecified unseen words from the vocabulary, and is treated thereafter as a seen word. We remark that the MLS of (8) and, for suitable reference pmfs Q, the estimate of (10) are well defined even for an infinite vocabulary, and we plan to investigate this more aesthetic alternative in the future.

To measure the efficacy of a pmf estimate, we compute the average codeword length (in bits) that the estimate $\hat{P}$ permits to achieve on the type $\hat{P}_{\rm Test}$ of the test set:

$$\frac{1}{N_{\rm Test}} \sum_{t=1}^{N_{\rm Test}} \log \frac{1}{\hat{P}(w_t)} = D(\hat{P}_{\rm Test}\|\hat{P}) + H(\hat{P}_{\rm Test}), \tag{16}$$

where $N_{\rm Test}$ is the size of the test set $\{w_1, \ldots, w_{N_{\rm Test}}\}$, and H the Shannon entropy.

4.1 Empirical Results for Unigram Estimation

We use the counts from Sections 00-22 to compute V. The procedure described above yields V=52743. The average codeword length (16) of the test set, along with the standard deviation based on the 10 smaller training sets, are shown in Table 1 for various pmf estimates.

- The Bayes estimate for $\alpha = 1$ is indistinguishable from $\hat{P}_\mu^*$ with a uniform Q.
- The use of a Zipf's reference pmf significantly improves over the uniform Q, and makes $\hat{P}_\mu^*$ almost

competitive with $\hat{P}_{\mathrm{GT}}$.

- The popularity of the Good-Turing estimator is evident from its low codeword length. Though not seen here, $\hat{P}_{\mathrm{GT}}(v)$ is not guaranteed to be monotonic in c_v.
- Using $\hat{P}_{\mathrm{GT}}$ as the reference pmf Q yields an estimate that is consistent with observed counts (cf property 7. in Section 2.), with no loss in codeword length. (The Good-Turing estimate was not inside the MLS in any of our experiments.)

The last column suggests that the MLS contains very competitive pmfs that may be found by using the best competitor as a reference, and getting other desirable properties for free.

4.2 Empirical Results for Bigram Estimation

We report preliminary results for bigram estimation in Table 2. Note that $V = 2,781,824,049$, the square of the unigram vocabulary size, and we estimate the joint distribution $\hat{P}(u, v)$ for consecutive words (u, v), not the conditional distribution $\hat{P}(v|u)$ of a word given the previous one. We collect bigram counts c_{uv} from Sections 00-22, compute the Good-Turing estimates of (14), and MLS-bases estimates with different reference Q's.

- It is clear that the Bayes estimate is dramatically poor, primarily due to the large vocabulary and a prior ($\alpha = 1$) that results in over-smoothing.
- Zipf's law as a reference Q improves $\hat{P}^*_\mu$ dramatically over the Bayesian estimate.
- $\hat{P}^*_\mu$ improves further if the Good-Turing estimate is used as a reference.

The MLS-based estimates $\hat{P}^*_\mu$ are competitive with the state of the art, and do not suffer from any nonmonotonicity.

5. Concluding Remarks

We have outlined a new way of viewing the problem of pmf estimation, particularly from small samples commonly encountered in numerous applications. The view, based on the notion of the maximum likelihood set defined in (8), opens many avenues of investigations not only in language modeling but in other areas of statistical estimation. The preliminary experiments presented here demonstrate that pmfs from the MLS selected using the K-L divergence criterion of (10) are theoretically well motivated, have many desirable properties and competitive performance (even in an application that has been studied for decades).

Readers interested in language modeling may note that comparisons are currently underway with other well known alternatives to Good-Turing [3]. Many of these alternatives, such as the Katz back off model [14], focus on *conditional* pmfs, e.g. $P(v|u)$. The MLS idea extends straightforwardly to this situation by estimating either (i) all conditional pmfs from a single bigram estimate of Section 4.2, or (ii) each conditional pmf $P(\cdot|u)$ separately, with a unigram estimate from Section 4.1 as the reference Q. The latter is particularly elegant, since $\hat{P}^*_\mu(\cdot|u)$ *naturally* reverts to the lower-order reference Q for unseen conditioning events u, where $N = 0$, and back off need not be externally enforced.

Acknowledgments

This research was partially supported by the National Science Foundation via Grant No ITR-0225656 and IIS-9982329, ARO DAAD19/-02-1-0337 and general funds from the Center for Imaging Science at The Johns Hopkins University.

References

[1] D. Wolpert and D. Wolf. Estimating functions of probability distributions from a finite set of samples. *Physical Review E*, 52:6841–6854, 1995.

[2] Frederick Jelinek. *Statistical Methods for Speech Recognition*. The MIT Press, 1998.

[3] S. F. Chen and J. Goodman. An empirical study of smoothing techniques for language modeling. In *Proceedings of the 34th Annual Meeting of the ACL*, pages 310–318, 1996.

[4] S. Khudanpur. A method of ME estimation with relaxed constraints. In *Proceedings of the Johns Hopkins University Language Modeling Workshop*, pages 1–17, 1995.

[5] J. Kazama and J. Tsujii. Evaluation and extension of maximum entropy models with inequality constraints. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, pages 137–144, 2003.

[6] S. F. Chen and R. Rosenfeld. A survey of smoothing techniques for ME models. *IEEE Transactions on Speech and Audio Processing*, 8(1):37–50, January 2000.

[7] B. Jedynak and S. Khudanpur. Maximum likelihood set for estimating a probability mass function. *Neural Computation*, 17(7):1508–1530, July 2005.

[8] I. Csiszar and J. Korner. *Information Theory: Coding Theorems for Discrete Memoryless Systems*. Academic Press, New York, 1981.

[9] I. Csiszar. I-divergence geometry of probability distributions and minimization problems. *The Annals of Probability*, 3(1):146–158, February 1975.

	$\hat{P}_{\text{Bayes}}$	$\hat{P}^*_\mu$: Unif-Q	$\hat{P}^*_\mu$: Zipf-Q	$\hat{P}_{\text{GT}}$	$\hat{P}^*_\mu$: GT-Q
Codeword Length $N \approx 100\text{K} \times 10$	10.57 ± 0.02	10.58 ± 0.02	10.40 ± 0.02	10.36 ± 0.02	10.37 ± 0.02
$P(v)$ for $c_v = 0$	6.53e-6 ± 0.06e-6	6.55e-6 ± 0.06e-6	2.00e-6 ± 0.04e-6	1.28e-6 ± 0.06e-6	1.28e-6 ± 0.06e-6
$P(v)$ for $c_v = 1$	1.30e-5 ± 0.01e-5	1.31e-5 ± 0.01e-5	1.55e-5 ± 0.07e-5	0.63e-5 ± 0.02e-5	0.93e-5 ± 0.01e-5
$P(v)$ for $\max c_v$	0.036 ± 0.001	0.036 ± 0.001	0.046 ± 0.001	0.054 ± 0.002	0.051 ± 0.002
Codeword Length $N \approx 1$ million	10.21	10.21	10.20	10.19	10.19

Table 1: Average codeword length (in bits) for unigram estimation, and the probability assigned to the most frequent and rare events, by various estimates: $\hat{P}_{\text{Bayes}}$ is the Bayes estimate (2) with $\alpha = 1$, $\hat{P}_{\text{GT}}$ is the Good-Turing estimate (11), $\hat{P}^*_\mu$ is the MLS estimate (10) with the reference pmf Q as indicated. Standard deviations (denoted $\pm$) are based on 10 different training samples against the same test set.

Sample size	$\hat{P}_{\text{Bayes}}$	$\hat{P}^*_\mu$: Zipf-Q	$\hat{P}_{\text{GT}}$	$\hat{P}^*_\mu$: GT-Q
$N \approx 100\text{K} \times 10$	15.09 ± 0.02	10.26 ± 0.06	8.87 ± 0.08	8.87 ± 0.08

Table 2: Average codeword length (in bits) for bigram estimation. Standard deviations (denoted $\pm$) are based on 10 different training samples against the same test set.

[10] I. J. Good. The populations frequencies of species and the estimation of population parameters. *Biometrika*, 40:237–264, 1953.

[11] W. Li. On-line references on Zipf's law, 1999. http://linkage.rockefeller.edu/wli/zipf/.

[12] L. Q. Ha, E. Sicilia, J. Ming, and F. J. Smith. Extension of Zipf's law to words and phrases. In *International Conference on Computational Linguistics (COLING'2002)*, pages 315–320, Taipei, Taiwan, Aug 2002 2002.

[13] C. T. Lawrence, J. L. Zhou, and A. L. Tits. User's guide for CFSQP version 2.5: A C code for solving (large scale) constrained nonlinear (minimax) optimization problems, generating iterates satisfying all inequality constraints. Technical Report TR-94-16r1, Institute for Systems Research, University of Maryland, 1997.

[14] S. Katz. Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Trans. Acoustics, Speech and Signal Processing*, 35(3):400–01, 1987.