

RESEARCH STATEMENT—DETAILED VERSION

We live amidst complex systems such as the Internet as well as advances like speech recognition, Mars missions and genome mapping, made possible by the unprecedented communication, computation, and storage resources available today. Modern research attempts to further these advances, and simultaneously harness available knowledge and data better.

Two aspects of this effort stand out.

First—while a lot of statistical work has been answering questions on sequences of symbols much longer than the set of possible symbols, the *alphabet size*—a number of current research problems require solutions for the opposite scenario, the large alphabet setting.

For instance, take speech recognition—language models for speech recognition estimate distributions over English words using text examples much smaller than the vocabulary. Or modern medicine—thousands of genes are clustered by their expression levels for applications in diagnosis and drug response prediction.

Tweaking old results and using them for large alphabets yields unsatisfactory results in many cases, and we are therefore forced to rework these problems altogether.

Second, problems posed by different systems are interconnected. But progress among interconnected problems do not always match up, typically due to lack of communication between the relevant fields.

For example, consider text compression on the one hand, along with language modeling for speech recognition which requires estimation of word probabilities given a text sample. It is folklore that compression and estimation are closely linked, but some very commonly used probability estimators in the machine learning community had not even been considered from a compression perspective till recently [1].

The theoretical focus of this statement will be the intersection of statistical learning and information theory. In particular, we concentrate on probability estimation, inference, classification and clustering in large alphabet settings. Practical applications include text classification and compression, image segmentation, analysis of gene expression data, and risk analysis for network insurance policies.

Many principles behind the approaches we take for the above problems are not just new and theoretically interesting, but have proved to be general in nature. As a demonstration, text classifiers based on these principles are comparable to the state of the art results (and faster) on several real life benchmarks using nothing but simple minded implementations of our algorithms, no tuning, no tweaking.

We start with some background on these problems, starting with compression and probability estimation.

1 Background

1.1 Universal data compression and large alphabets

While Huffman or Shannon codes achieve compression of data by using the underlying distribution to assign variable length codewords, in most applications, we do not know the underlying distribution. In such situations, the source is usually modeled as an unknown distribution in a collection \mathcal{P} of distributions, *e.g.*, the collection of *i.i.d.*, markov, context tree sources, depending on the context of the problem.

The question is: how much penalty is incurred if we use one (albeit carefully chosen) encoder for the whole class \mathcal{P} instead of a separate one for each distribution? This penalty is measured by the *redundancy* of encoding.

However, Kieffer showed that universal compression of even *i.i.d.* sources over infinite alphabets incurs infinite redundancy [2]. (We analyze this problem in more detail in [3].)

1.2 Estimation with insufficient data

Probability estimation over large alphabets is complicated owing to the large number of rare and unseen events whose probabilities cannot be reliably estimated using conventional maximum likelihood approaches. To resolve this difficulty, Good and Turing [4] derived an unintuitive, yet effective, formula for estimating the probability of every symbol appearing in a sequence, as well as the probability of the missing mass. Note that Good and Turing’s formula does not attempt to assign a probability to every missing element, it only estimates the probability of the set of all missing elements.

Several estimators based on this formula have since been employed in a variety of applications such as information retrieval, spelling correction, and language modeling. These *Good-Turing* estimators are known to work very well for rare events. However, while there were justifications [5] for why the estimator should work well, until recently [6] there was not much theoretical analysis on these estimators.

Not only are these estimators a little bit of black magic in the sense that their underlying principles are not always completely understood, they do not seem to always work better in every application. In fact, certain heuristics outperform these estimators in tasks like text classification (naive bayes approaches).

2 New theory

We therefore need new solutions in both problems above. To proceed further we observe that estimation of the probability distribution in the large alphabet setting is a futile task—Kieffer’s result mentioned in Section 1.1 can be used to show this formally. Therefore, what we try to estimate is the probability of observing just those events already in the data sample.

In the compression framework, this can be seen to be equivalent to separating [7] the description of the data strings over large alphabets into two parts: description of the symbols appearing in the sequence, and of their *pattern*, the order in which the symbols appear. For instance, to describe the sequence “Sam I am that Sam I am I do not like that Sam I am” over the symbol space of English words, this approach separates the pattern, 123412325674123 from the dictionary,

1	2	3	4	5	6	7
Sam	I	am	that	do	not	like

Patterns have an interesting combinatorial structure [3], and using Hardy and Ramanujan’s celebrated results on the number of integer partitions [8, 9], we show that patterns of *i.i.d.* sequences **can** be universally compressed with vanishing redundancy [7], see also [10]. These results have since been extended to more complex distributions as well [11].

These results have their analog in estimation as well. We also analyze a few variants of the Good Turing estimator, showing that while they outperform other commonly used estimators, none is optimal in general. We subsequently develop two provably asymptotically optimal estimators. These results appeared in Science [12].

3 New directions

3.1 Distribution modeling and computation of matrix permanents

In [13], we consider the problem of *distribution modeling*, where we estimate the values of the probabilities in the distribution, but not their mapping to the support. For example, we would estimate from the data that the underlying distribution contains 2 symbols with probabilities 2/3 and 1/3, but not which symbol has probability 2/3. These estimates of distribution are useful in a wide variety of applications, *e.g.*, in the computation of entropy of the distribution.

From the pattern of the data, we use the Expectation-Maximization algorithm and Markov chain Monte Carlo sampling heuristics to model the distribution. This approach seems to work well even when the data sample is too small for conventional techniques [14]. Consistency of this estimate is shown in [15].

Theoretically, the Markov chain sampling approximations are related to the estimation of the *permanent* of a matrix, a problem that is #P-complete [16]. Computation of the permanent is a well studied problem as well, and it arises in the contexts of estimating the number of bipartite matchings, computation of network reliability, etc. While polynomial time algorithms to compute the permanent have been obtained [17], they are not fast in practice. We are working on improving existing results for certain subsets of matrices.

3.2 Text compression and language models

In text compression and language modeling, the description of the dictionary [18] is empirically known to contribute very little relative to the number of bits needed to specify the pattern. Hence this application essentially reduce to compressing the patterns of sequences well. The results outlined above are the basis of an ongoing effort to build language models that outperform currently used ones.

3.3 Support independent statistics

Support independent statistics do not depend on the values of elements in the support of the distribution that is sampled from. Examples include entropy and support size, while examples of statistics that are not support independent include the mean of the distribution.

There are a wide range of problems from caching to chip testing that involve estimating the support independent statistics of a set which can be tackled by the distribution modeling techniques described in Section 1.2.

3.3.1 Estimation of support size

Application areas: databases, entropy estimation, caching, chip testing

Pattern based estimators seem to approximate the distribution well even when the sample size is comparable to the alphabet. It is natural to ask how many samples are necessary to estimate the distribution to a given accuracy. While there can be no general bounds [19], what happens in special cases, for instance, Zipf distributions with bounded alphabet size? In particular these questions are relevant for web-caching. How do we best use these estimators for caching applications? A preliminary set of results was published in [20]

3.3.2 Estimating information theoretic quantities

Application areas: classification, biology

While estimation of entropy or mutual information is not identical to estimation of the underlying distribution, a good approach to the distribution estimation problem does help, and is currently the best approach known under some conditions [21]. Results from [13] show promise in using a plug-in estimator based on distribution modeling from patterns.

3.4 Classification

In [22], we adapted some of these techniques to implement a text classification scheme that compares favorably with the current state of the art classification schemes. (See Table 1)

The collection *Newsgroups* is a list of 1000 articles collected from 20 newsgroups. Among the newsgroups are collections of closely related newsgroups such as `comp.os.ms-windows.misc`, `comp.windows.x` or `rec.autos` and `rec.motorcycles`. The task is to identify all the newsgroups. Roughly 10% of the documents were randomly chosen for training, while the rest are used for testing, and the results confirmed by repeat trials over random splits. CADE is a Portugese dataset with 12 classes. The improvements in classification accuracy are statistically significant at better than .05 level.

We are currently combining the approach above with graphical model inference techniques to yield a richer class of classifiers. For details, please contact me at `prasadsn@eecs.berkeley.edu`.

Database	Newsgroups	CADE
SVM	73.09	52.84
New	73.46	59.24

Table 1: Percentage of documents accurately classified. (SVM: support vector machines)

3.4.1 Semi supervised classification

Typical statistical approaches for classification learn using labelled data, namely the class label is provided for the training data used.

While it is expensive to create labeled training data, it is usually easier to come up with lot of unlabeled training material—therefore, the question is, can we also use the unlabeled training data?

On the face of it, unlabeled training material is useless. Remarkably, in some applications including text classification, the structure of the classification problem actually allows us to improve classification. In the naive bayes setting, we use unlabeled data by predicting their label using the labeled data and subsequently using these predicted labels to train the classifier.

Of course, such approaches cannot always be succesful. We need to know when we can actually trust our predicted labels to improve classification. To answer these questions better, we currently are developing a framework to analyze how large alphabet prediction schemes perform.

3.5 Clustering

Clustering problems crop up in data mining, image segmentation, biology (MRI image segmentation), and bioinformatics. Many of these problems require us to cluster a set of large alphabet sequences by observing a subset of the sequences. Typically, the large alphabet renders practical sizes of this subset to be very small.

Example : For disease diagnosis using protein profiling *e.g.*, [23], relevant cell samples are first collected from several subjects. Using mass spectrometric techniques, the proteins present in each sample, which could number several hundred or more, are determined. The samples are then clustered so as to distinguish diseased ones from normal ones. Since the number of subjects and samples available rarely even exceeds the number of proteins in most experiments, we are stuck with the familiar large alphabet issue—the alphabet (the set of proteins) renders the available input (the samples from various subjects) effectively small. \square

See [24] for several other large alphabet clustering problems.

Theoretically, we handle this as a constrained distribution estimation problem. In previous applications, we were concerned with learning a distribution on our sample space—here we impose an additional constraint on the distributions—for example, the distribution has to be a mixture of gaussians, a mixture of binomials, etc.—the challenge being to incorporate this constraint into our large alphabet approaches.

4 Conclusion

The host of problems identified above attempt to integrate novel information theory into large alphabet (high dimensional) problems already encountered in many modern problems. In the interest of brevity, only a subset of work in progress has been outlined.

Among the topics left out from this statement is a proposal for risk analysis to insure networks. For more details, please contact me at prasadsn@eecs.berkeley.edu.

The proof of any applied science is in the results, and in this regard, our optimism is justified by the ease with which the principles above perform well in practical problems (text classification). Last but not the least is that we can also make this theoretically interesting—part of this work [7] won the 2006 Information Theory society best paper award.

References

- [1] A. Orlitsky, N.P. Santhanam, and J. Zhang. Always Good Turing: Asymptotically optimal probability estimation. In *Proceedings of the 44th Annual Symposium on Foundations of Computer Science*, October 2003.
- [2] J.C. Kieffer. A unified approach to weak universal source coding. *IEEE Transactions on Information Theory*, 24(6):674–682, November 1978.
- [3] A. Orlitsky and N.P. Santhanam. Speaking of infinity. *IEEE Transactions on Information Theory*, 50(10):2215–2230, October 2004.
- [4] I.J. Good. The population frequencies of species and the estimation of population parameters. *Biometrika*, 40(3/4):237–264, December 1953.
- [5] A. Nadas. On Turing’s formula for word probabilities. *IEEE Transactions on Acoustics, Speech and Signal Processing*, ASSP-33(6):1414–1416, December 1985.
- [6] D. McAllester and R. Schapire. On the convergence rate of Good Turing estimators. In *Proceedings of the Thirteenth Annual Conference on Computational Learning Theory*, 2000.
- [7] A. Orlitsky, N.P. Santhanam, and J. Zhang. Universal compression of memoryless sources over unknown alphabets. *IEEE Transactions on Information Theory*, 50(7):1469–1481, July 2004.
- [8] G.H. Hardy and S. Ramanujan. Asymptotic formulae in combinatory analysis. *Proceedings of London Mathematics Society*, 17(2):75–115, 1918.
- [9] J.H. van Lint and R.M. Wilson. *A course in combinatorics*. Cambridge University Press, 1996.
- [10] N. Jevtić, A. Orlitsky, and N.P. Santhanam. A lower bound on compression of unknown alphabets. *Theoretical Computer Science*, Feb 2005.
- [11] A. K. Dhulipala and A. Orlitsky. Bounds on the redundancy of HMM patterns. Submitted to 2004 Proceedings of IEEE Symposium on Information Theory.
- [12] A. Orlitsky, N.P. Santhanam, and J. Zhang. Always Good Turing: Asymptotically optimal probability estimation. *Science*, 302(5644):427–431, October 17 2003.
- [13] A. Orlitsky, N.P. Santhanam, K. Viswanathan, and J.Zhang. On modeling profiles instead of values. In *Uncertainty in Artificial Intelligence*, 2004.
- [14] A. Orlitsky, Sajama, N.P. Santhanam, K. Viswanathan, and J. Zhang. Practical algorithms for modeling sparse data. Proceedings of the 2004 Proceedings of IEEE Symposium on Information Theory.
- [15] A. Orlitsky, N.P. Santhanam, K. Viswanathan, and J. Zhang. Convergence of profile based estimators. In *Proceedings of the IEEE Symposium on Information Theory*, 2005.
- [16] L.G.Valiant. The complexity of computing the permanent. *Theoretical Computer Science*, 8:189–201, 1979.
- [17] M. Jerrum, A.Sinclair, and Vigoda. A polynomial-time approximation algorithm for the permanent of a matrix with non-negative entries. *Theory of computing*, pages 712–721, 2001.
- [18] A. Dhulipala and A. Orlitsky. Experiments with text. In preparation.
- [19] A. Orlitsky, N.P. Santhanam, K. Viswanathan, and J. Zhang. Innovation and pattern entropy of stationary processes. In *Proceedings of the IEEE Symposium on Information Theory*, 2005.
- [20] A. Orlitsky, N.P. Santhanam, and K. Viswanathan. Population estimation with performance guarantees. In *Proceedings of IEEE Symposium on Information Theory*, 2007.

- [21] A. Wyner and D. Foster. On the lower limits of entropy estimation. Submitted to IEEE Transactions on Information Theory.
- [22] N.P. Santhanam, A. Orlitsky, and K. Viswanathan. New tricks for old dogs: Large alphabet probability estimation. In *Information Theory Workshop (invited)*, Lake Tahoe, CA, 2007.
- [23] H. Bensmail, B. Aruna, O.J. Semmes, and A. Haoudi. Functional clustering algorithm for high-dimensional proteomics data. *J. Biomed. Biotechnol.*, v2005(2):80—86, 2005.
- [24] *Workshop on clustering high dimensional data and its applications*, Lake Buena Vista, Florida, April 2004. Held in conjunction with Fourth SIAM International conference on data mining.