

Nikola Jevtić

RESEARCH STATEMENT

During my graduate school, I participated in the development of large vocabulary continuous speech recognition system, together with Aldebaro Klautau and our advisor Alon Orlitsky. We felt that there are many parts of the speech recognition system not well understood or justified, but because of the fine tuning they received over the decades of development had performance very hard to surpass. We examined those choices with modern machine learning tools and ideas, and with models that can offer strong mathematical performance guarantees. Consequently my research spanned many fields such as information theory, signal processing, statistical learning theory, and algorithms. The system started as a small vocabulary recognizer, as in [1] where we demonstrated a Java applet for voice web navigation. By June 2004, we will have a full large vocabulary decoder. That will not be the end of the road, but just the beginning as it would provide an excellent testbed for novel machine learning theories.

The following paragraphs describe my prior research work as well as my current and future research interests.

Universal compression and language modeling [2, 3, 4]

In virtually all practical large vocabulary systems, the deployed language models are based on the *n-gram* approximation where a distribution is estimated for every word following a sequence of previous $n - 1$ words. In our research we investigated the way the distribution parameters are estimated. The major problem for probability estimation is the data sparsity. The current methods use ad-hoc smoothing techniques to avoid assigning zero probability to any sentence, while staying close to empirical frequencies in the training data. We evaluated some of those smoothing methods from the aspect of the universal compression theory.

In universal compression, the goal is to compress unknown sources essentially at their entropy rate. The only available information is that the source belongs to a certain (possibly parametric) class of sources, such as *i.i.d.*, or Markov. It is not always possible to efficiently learn the source, and we looked at two distinct scenarios. *The finite alphabet* case considers the source over alphabet that is finite and known in advance, and *the unknown alphabet* case assumes no knowledge of the source alphabet, which might even be infinite.

The finite alphabet case has been analyzed in the literature and many results exist regarding the optimal learning rates and the asymptotically optimal methods achieving them. We demonstrated [3] how analytic results match the empirical results, when comparing several simple smoothing methods. We further experimented with smoothing techniques trying to learn the optimal distributions for n-gram models and designed a slightly better model (model that assigns higher probabilities to the unseen text). More importantly, the new model suggests the existence of words whose counts would grow slower than linearly in the increasing sample, and that hurt the language model performance. This intuitive result illustrating nonlinear (or maybe nonstationary) nature of the language suggests to try to identify such words (on Wall Street Journal(WSJ) database that we used in our experiments those might be some temporal topics that never reoccur) and to eliminate them from the training data before building the final model. Until now, the conventional wisdom with language modeling was to add more data to the training in order to improve the model. However, our results suggest that adding more data indiscriminately, would introduce more of the words with sublinear growth, and thus slow down the convergence to the optimal distribution.

The unknown alphabet case seemed more difficult since Kieffer showed that universal compression of unknown alphabet sources is not possible, *i.e.* in the worst case the redundancy is infinite. It turns out that the problem can be split in two parts, the description of the dictionary, and the description of the word pattern. Further, it can be shown that the pattern can be efficiently compressed and thus the difficulty lies only in the dictionary [2]. For language modeling we do not need to encode the dictionary to learn the distribution. It is sufficient just to determine how likely it is to see again

the word that was previously observed certain number of times. Orłitsky et al. showed that a variant of Good-Turing method is universal for the set of sources over unknown alphabet with redundancy of $\mathcal{O}(n^{\frac{2}{3}})$ bits per block of n samples, and that the optimal universal code has redundancy no bigger than $\mathcal{O}(n^{\frac{1}{2}})$. Together with Narayana Santhanam and Alon Orłitsky, we showed that the optimal code must have redundancy of at least $\mathcal{O}(n^{\frac{1}{3}})$ [4]. There are many open problems in this setting, and most important are determining the redundancy of the optimal code, and finding the practical code that could perform close to the optimal.

Estimated rank pruning [5, 6]

Most decoders use the so called beam search to find the most likely spoken sentence. At every time frame a list of active hypotheses is compiled, and all the hypotheses with scores worse by a prespecified amount from the best hypothesis are discarded. Sometimes the number of hypotheses falling within the beam may be too high to handle, either because of time or memory constraints (*e.g.*, if the recognizer must run in real time), and a maximum number of allowable hypotheses for each time frame is specified. The standard implementation called *histogram pruning* computes the histogram at each time frame of all the hypotheses and discards the extra ones based on the histogram cutoff closest to the desired maximum number. We demonstrated that the number of hypotheses falling inside the beam grows exponentially with the beam size. This allowed us to use an estimated threshold to prune hypotheses while they are generated and not accumulate all until the end of the frame. When the last hypothesis is processed, the beam estimate can be adjusted to the more accurate value based on the exponential interpolation through two points. This method, the *estimated rank pruning* [5] shows no loss in recognition accuracy compared to the exact sorting implementation, although it is possible to lose some of the valid hypotheses if the initial estimate is incorrect. The advantage of the estimated rank pruning over the histogram pruning is that many of the hypotheses do not need to be preserved until the frame ends, but can be discarded, and resources can be released. One could not use this pre-pruning stage with the histogram pruning, because when overpruning occurs, histogram cannot show where the pruning threshold should have been, and it would be impossible to compensate for the estimate in the next frame.

In [6] we showed that our estimate is good even for small targets, when we reversed the problem and ran a recognizer with a very tight pruning and required at least a minimum number of active hypotheses.

Efficient large vocabulary decoder

The decoder's task is to efficiently combine all the available knowledge sources to produce the most likely word sequence. Typical knowledge sources are the language, pronunciation and acoustic models. There are many architectures and approaches used in large vocabulary decoders, including multiple passes and stack algorithms. However, for most tasks a single pass synchronous beam search is deployed, and there are two main flavors.

The Finite State Transducer approach advocates compiling all the knowledge sources into a single network optimized for size and fast execution. This significantly simplifies the decoder, while combining equivalent paths minimizes the redundant computations in the search stage. The drawback is that the composition algorithm is expensive to execute, and can only be done offline, on a specialized hardware. This prevents any online adaptation to the language or pronunciation models, which might be the key for achieving true speaker- and topic-free recognition.

The competing approach suggests dynamic generation only of those parts of recognition network that are needed for the decoding. It allows easier adaptation to the changing environment at the expense of the additional processing in the recognition stage. The complexity of the partial network generation depends on the models used for the recognition. If cross-word context dependent models are used, both

the size of the generated block and the time needed to create it increase dramatically.

In our decoder, we build on the approach taken by Nguyen et al. They proposed the *Ewaves* algorithm for updating the active hypotheses. It is based on the observation that if a grammar has a tree form, then unique ordering can be specified, and all the hypotheses can be visited in this fixed order. This eliminates the need to maintain two lists of active hypotheses, one for the current frame and another for the hypotheses reaching the next frame. These lists are usually huge hash tables with 10 to 100 thousand hypotheses. Instead, all hypotheses can be kept in a single linked list and updated in place. The advantages of this approach are the savings of space (having just a single list), and time (having to look up pointer for the next hypothesis instead of retrieving a hypothesis from a hash table). In their implementation, they did not use cross-word dependency for acoustic models to maintain the tree structure inside every language model context. In our approach, we decided to create phonetic context dynamically, by collecting the phonetic labels from the traversed path in the monophone network. Thus for every edge in the network we maintain a list of active models, as there may be more than a single path that reached it. The benefits are that the network is much smaller, representing just the monophone models, always in the tree form, and much easier to create on demand, even partially. It also holds promise for running true pentaphone decoding. Normally the network size goes beyond usable when pentaphones are used and the standard trick is to use just triphone cross-word context. We believe that during the actual recognition process, just a tiny fraction of the possible models would be active, and that there would not be any additional tax on the system. In the preliminary tests, our system (written in Java) was 14 times faster than the Mississippi State University prototype system (C++). We used only twice the memory footprint on the WSJ bigram 5000 words vocabulary with fully expanded (but not minimized in order to keep the tree structure) language model.

In order to run longer span language models with larger vocabularies, efficient dynamic tree generation needs to be added to the system. It will then offer tools for quick hypotheses testing regarding various front ends, acoustic and language models on the relevant databases for current speech research.

On alternatives to Gaussian mixture models and HMMs [7, 8, 9]

In our search for the acoustic models alternative to Gaussian mixtures, we looked at the Support Vector Machines. They offer binary classifiers that outperform other techniques on many machine learning tasks. The first concern was that the problem at hand was not binary but multiclass, and we researched the ways of combining binary classifiers to obtain multiclass equivalents. Together with fellow student Aldebaro Klautau and our advisor we improved theoretical generalization bounds for combining binary classifiers and showed both theoretically and practically that all-pairs Error Correcting Output Code (ECOC) matrix outperforms the more commonly used one-against-all ECOC matrix [7, 8].

In the follow-up we tried to compare several kernel based methods (Support Vector Machine(SVM), Relevance Vector Machine(RVM), Informative Vector Machine(IVM)) with discriminatively trained Gaussian Mixture Models(DGMM). We compared them on several machine learning tasks, not all of them related to speech. The results indicated that when the underlying source has distribution close to Gaussian, DGMM outperform kernel methods [9]. This may suggest that discriminative training of matched distributions should be preferred whenever the underlying process that we model is well understood. This makes a prospect for using kernel methods like SVM, RVM and IVM for speech recognition less likely. At least it seems that DGMM will not be outright replaced by kernel methods, although we do not exclude some kind of symbiosis.

Future research objectives

When the decoder is finished, it will allow testing the new paradigms on relevant large vocabulary tasks in competitive runtime. There are several major directions I find important. For language models, an interesting research objective is finding theoretically optimal smoothing methods for the case of unknown vocabularies, as well as identifying the words with sub-linear growth.

Another theoretical direction, not focused exclusively on speech, would be to bring universal compression theory to continuous distributions like Gaussians. People have used Bayesian priors for learning Gaussian distributions in the context of incorporating the prior knowledge or for smoothing, but there are no theoretical guarantees on the learning speed or results showing preference for some priors against others. The solution to the game theoretic question similar to the one for discrete distributions would increase our understanding of learning, and facilitate better results with less training data. Practical solutions would be relevant to a broad set of applications.

Another interesting avenue to explore would be to combine kernel methods with HMMs, and try to use them as additional information when the system does not trust the Gaussian mixtures. This could lead to systems more robust to noisy environments.

Finally, the recognition system can be used to experiment with various signal processing techniques for noise canceling and better, more robust front ends to improve recognition in challenging environments.

Another area interesting to me is the theory and application of graphical models. Those methods have found broad range of successful applications in many areas including coding, signal processing, speech recognition, bioinformatics, etc. I would be interested in extending both the theoretical boundaries and designing the new applications, as well as improving algorithms related to the existing applications. New applications are likely to emerge in fields like bioinformatics and bioengineering. Those fields hold great promise for quality improvement of our lives if we manage to extract the necessary information from the data, and cures for many difficult diseases could be found.

Finally, I understand that a good research program is not possible without adequate financial support, and I intend to actively seek research sponsorship from both governmental and industrial sources.

References

- [1] Aldebaro Klautau, Nikola Jevtić, and Alon Orlitsky. Server-assisted speech recognition over the internet. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2000.
- [2] Nikola Jevtić, Alon Orlitsky, and Narayana Santhanam. Universal compression of unknown alphabets. In *Proceedings of IEEE Symposium on Information Theory*, 2002.
- [3] Nikola Jevtić and Alon Orlitsky. On the relation between additive smoothing and universal coding. In *Automatic Speech Recognition and Understanding Workshop*, 2003.
- [4] Nikola Jevtić, Alon Orlitsky, and Narayana Santhanam. A lower bound on compression of unknown alphabets. In preparation.
- [5] Nikola Jevtić, Aldebaro Klautau, and Alon Orlitsky. Estimated rank pruning and Java-based speech recognition. In *Automatic Speech Recognition and Understanding Workshop*, 2001.
- [6] Nikola Jevtić, Aldebaro Klautau, and Alon Orlitsky. Estimated rank pruning for speech recognition. unpublished, <http://cwc.ucsd.edu/~nikola/research.html>.
- [7] Aldebaro Klautau, Nikola Jevtić, and Alon Orlitsky. Combined binary classifiers with application to speech recognition. In *International Conference on Spoken Language Processing*, 2002.
- [8] Aldebaro Klautau, Nikola Jevtić, and Alon Orlitsky. On nearest-neighbor ECOC with application to all-pairs multiclass SVM. *Journal of Machine Learning Research*, 4:1–15, April 2003.
- [9] Aldebaro Klautau, Nikola Jevtić, and Alon Orlitsky. Discriminative gaussian mixture models: A comparison with kernel classifiers. In *International Conference on Machine Learning*, 2003.