Interpolated Spectral NGram Language Models

Ariadna Quattoni and **Xavier Carreras**
dMetrics
Brooklyn, NY 11211
{ariadna.quattoni,xavier.carreras}@dmetrics.com

Abstract

Spectral models for learning weighted non-deterministic automata have nice theoretical and algorithmic properties. Despite this, it has been challenging to obtain competitive results in language modeling tasks, for two main reasons. First, in order to capture long-range dependencies of the data, the method must use statistics from long substrings, which results in very large matrices that are difficult to decompose. The second is that the loss function behind spectral learning, based on moment matching, differs from the probabilistic metrics used to evaluate language models. In this work we employ a technique for scaling up spectral learning, and use interpolated predictions that are optimized to maximize perplexity. Our experiments in character-based language modeling show that our method matches the performance of state-of-the-art ngram models, while being very fast to train.

1 Introduction

In the recent years we have witnessed the development of spectral methods based on matrix decompositions to learn Probabilistic Non-deterministic Finite Automata (PNFA) and related models (Hsu et al., 2009, 2012; Bailly et al., 2009; Balle et al., 2011; Cohen et al., 2012; Balle et al., 2014). Essentially, PNFA can be regarded as recurrent neural networks where the function that predicts the dynamic state representation from previous states is linear. Despite the expressiveness of PNFA and the strong theoretical properties of spectral learning algorithms, it has been challenging to get competitive results on language modeling tasks. We argue and confirm with our experiments that there are two main reasons why using spectral methods for language modeling is challenging. The first reason is a scalability problem to handle long range dependencies. The spectral method is based on computing a Hankel matrix that contains statistics of expectations over substrings generated by the target language. If we want to incorporate long-range dependencies we need to consider long substrings. A consequence of this is that the Hankel matrix can become too large to make it practical to perform algebraic decompositions. To address this problem we use the basis selection technique by Quattoni et al. (2017) to scale spectral learning and model long range dependencies. Our experiments confirm that modeling long range dependencies is essential to obtain competitive language models.

The second limitation of classical spectral methods when applied to language modeling is that the loss function that the learning algorithm attempts to minimize is not aligned with the loss function that is used to evaluate model performance. Spectral methods minimize the ℓ_2 distance on the prediction of expectations of substrings up to a certain length (see Balle et al. (2012) for a formulation of spectral learning in terms of loss minimization), while language models are usually evaluated using conditional perplexity. There have been some proposals on generalizing the fundamental ideas of spectral learning to other loss functions (Parikh et al., 2014; Quattoni et al., 2014). However, while these approaches are promising they have the downside that they lead to relatively expensive iterative convex optimizations and it is still a challenge to scale them to model long-range dependencies.

In this paper we propose a simpler yet effective alternative to the iterative optimization. We use the classical spectral method based on low-rank matrix decomposition to learn a PNFA that computes substring expectations. Then we use these expectations as features in an interpolated ngram model and we learn the weights of the interpolation so as to maximize perplexity. This interpo-

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 5926–5930
Florence, Italy, July 28 - August 2, 2019.

lation step is iterative, but it is a simple and very efficient convex optimization: the weights of the interpolation can be trained in a few seconds or minutes at most. The refinement step allows us to leverage all the moments computed by the learned PNFA and to align the spectral method with the perplexity evaluation metric.

Our experiments on character-level language model show that: (1) modeling long range dependencies is important; and (2) with the simple interpolation step we can obtain competitive results. Our perplexity results are significantly better than feed-forward NNs, as good or better than sophisticated interpolation techniques such as Kneser-Ney estimation, and close to the performance of RNNs on two datasets.

The main contribution of our work consists on combining two simple ideas, i.e. incorporating long-range dependencies via basis selection of long substring moments (Section 2), and refining the predictions of the PNFA with an iterative interpolation step (Section 3). Our experiments show that these two simple ideas bring us one step closer to making spectral methods for PNFA reach state-of-the-art performance on language modeling tasks (Section 4). The advantage of these methods over other popular approaches to language modeling is their simplicity and the fact that they rely on efficient convex optimizations for training the model parameters. Furthermore, PNFA are probabilistic models for which efficient inference methods can be easily derived for computing all sorts of expectations. These expectations could then be used as features to learn predictive interpolation models. In this paper we present experiments with one type of expectation and interpolation model that illustrates the potential of this approach.

2 Spectral Language Models

2.1 Probabilistic Non-Deterministic Finite Automata

We start describing the general class of Weighted Automata over strings. Let $x = x_1 \cdots x_n$ be a sequence of length n over some finite alphabet Σ. We denote as $\Sigma^\star$ the set of all finite sequences, and we use it as a domain of our functions. We use $x \cdot x'$ to denote the concatenation of two strings x and x'.

A Non-Deterministic Weighted Automaton (WA) with k states is defined as a tuple: $A = \langle \boldsymbol{\alpha}_0, \boldsymbol{\alpha}_\infty, \{\mathbf{A}_\sigma\}_{\sigma\in\Sigma}\rangle$ with: $\boldsymbol{\alpha}_0$, $\boldsymbol{\alpha}_\infty \in \mathbb{R}^k$ are the initial and final weight vectors; and $A_\sigma \in \mathbb{R}^{k\times k}$ are the transition matrices associated to each symbol $\sigma \in \Sigma$. The function $f_A : \Sigma^\star \to \mathbb{R}$ realized by an WA A is defined as:

$$f_A(x) = \boldsymbol{\alpha}_0^\top \mathbf{A}_{x_1} \cdots \mathbf{A}_{x_n} \boldsymbol{\alpha}_\infty \quad . \qquad (1)$$

Probabilistic Non-Deterministic Finite Automata (PNFA) are WA that compute a probabilistic distribution over strings. One can easily transform a PNFA into another automata that computes substring expectations via simple transformations of the model parameters, and the reverse is also true, see Balle et al. (2014) for details. In this paper we will directly learn and use automata that compute expectations. With these expectations we will calculate the conditional probabilities of a language model[1]:

$$\Pr[\sigma \mid x_{1:n}] = \frac{f_A(x_{1:n} \cdot \sigma)}{\sum_{\sigma'\in\Sigma} f_A(x_{1:n} \cdot \sigma')} \qquad (2)$$

Here, n is the length of the left context, analogous to the order of an NGram model, but we compute the expectations not from counts but from a PNFA.

2.2 The Spectral Method

We now give a brief description of the spectral method for estimating a PNFA that computes expectations over substrings. We only provide a higher-level description of the method; for a complete derivation and the theory justifying the algorithm we refer the reader to the works by Hsu et al. (2009) and Balle et al. (2014).

Assume a distribution of strings over some discrete alphabet, our target function $f(x)$ is the expected number of times that x appears as a substring of a string sampled from the distribution. At training, we are given strings $\mathcal{T}$ from the distribution and we want to estimate f. We denote as $f_\mathcal{T}(x)$ the empirical substring expectation of x in $\mathcal{T}$.[2] Using $f_\mathcal{T}$, the spectral method estimates a WA A with k states, where k is a parameter of the algorithm, such that f_A is a good approximation of f. The method reduces the learning problem to computing an SVD decomposition of a special type of matrix called the Hankel matrix, that collects the observed expectations $f_\mathcal{T}$. The method is described by the following steps:

[1] For language models, we assume that Σ includes a special symbol for end of sentence.

[2] This corresponds to the number of times that x is observed as substring of any string in $\mathcal{T}$, normalized by the number of strings in $\mathcal{T}$.

(1) Select a set of prefixes $\mathcal{P}$ and suffixes $\mathcal{S}$, that will serve as indices of the Hankel matrix for rows and columns respectively. A typical choice is to select all substrings up to a certain size n, but this quickly grows, and in practice prior work uses a small n. Instead we use the basis selection technique presented by Quattoni et al. (2017), which allows to capture long-range dependencies (analogous to having a large n) but keeping the number of prefixes and suffixes manageable.

(2) Compute Hankel matrices for $(\mathcal{P}, \mathcal{S})$.

   (a) Compute $\mathbf{H} \in \mathbb{R}^{\mathcal{P} \times \mathcal{S}}$, with entries $\mathbf{H}(p, s) = f_{\mathcal{T}}(p \cdot s)$.

   (b) Compute $\mathbf{h}_{\mathcal{P}} \in \mathbb{R}^{\mathcal{P}}$ with $\mathbf{h}_{\mathcal{P}}(p) = f_{\mathcal{T}}(p)$ and $\mathbf{h}_{\mathcal{S}} \in \mathbb{R}^{\mathcal{S}}$ with $\mathbf{h}_{\mathcal{S}}(s) = f_{\mathcal{T}}(s)$.

   (c) For each $\sigma \in \Sigma$, compute $\mathbf{H}_\sigma \in \mathbb{R}^{\mathcal{P} \times \mathcal{S}}$ with entries $\mathbf{H}_\sigma(p, s) = f_{\mathcal{T}}(p \cdot \sigma \cdot s)$.

(3) Compute a k-rank factorization of $\mathbf{H}$. Compute the truncated SVD of $\mathbf{H}$, i.e. $\mathbf{H} \approx \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$ resulting in a matrix $\mathbf{F} = \mathbf{U}\mathbf{\Sigma} \in \mathbb{R}^{\mathcal{P} \times k}$ and a matrix $\mathbf{B} = \mathbf{V} \in \mathbb{R}^{\mathcal{S} \times k}$. Thus $\mathbf{H} \approx \mathbf{F}\mathbf{B}^\top$ is an k-rank factorization of $\mathbf{H}$.

(4) Recover the WA A of k states. Let $\mathbf{M}^+$ denote the Moore-Penrose pseudo-inverse of a matrix $\mathbf{M}$. The elements of A are recovered as follows. Initial vector: $\boldsymbol{\alpha}_0^\top = \mathbf{h}_{\mathcal{S}}^\top \mathbf{B}$. Final vector: $\boldsymbol{\alpha}_\infty = \mathbf{F}^+ \mathbf{h}_{\mathcal{P}}$. Transition Matrices: $\mathbf{A}_\sigma = \mathbf{F}^+ \mathbf{H}_\sigma \mathbf{B}$, for $\sigma \in \Sigma$.

The computation is dominated by step (3), the SVD of the Hankel matrix, which is at most cubic in the size of the matrix. In practice, this method is scalable and fast to train.

3 Interpolated Predictions

One limitation of the spectral method is that the loss that it minimizes is not aligned with the probabilistic metrics used in language modeling, such as perplexity. Instead the spectral method minimized the ℓ_2 loss over the observed empirical moments, i.e. those substrings collected in the Hankel matrix. To align the loss function with a perplexity measure we propose a simple refinement step, where we use the expected counts computed by the learned PNFA as features of a log-linear model, and learn interpolation weights. In contrast to Equation 2, which uses the longest context x of length n to compute the conditional probability, the interpolated model leverages the ability of the PNFA to model substring expectations of all lengths up to n. This is similar to classic interpolation of language models (Rosenfeld, 1994; Chen, 2009).

Given a function f computing substring expectations, the interpolation is:

$$g(x_{1:n}, \sigma) = \exp\left\{ \sum_{j=0}^{n-1} w_{\sigma,j} \log f(x_{n-j:n} \cdot \sigma) \right\} \quad (3)$$

where $x_{1:n}$ is a context of size n, σ is the output symbol, and $w_{\sigma,j}$ are the interpolation weights, with one parameter per output symbol σ and context length j, with $0 \leq j < n$.

As it is standard with interpolation models, we train the weights by maximizing the conditional log-likelihood of the development set. We assume that f is fixed, which results in a convex optimization, and we solve with L-BFGS.

4 Experiments

We present experiments in character-based language modeling. Our spectral ngram models work with a fixed context length, and we show results varying this length up to relatively large values.

Following the standard, the goal is to learn a language model that predicts the next symbol given a sentence prefix, including the prediction of sentence ends. As datasets we use the Penn Treebank (PTB) prepared by Mikolov et al. (2012)[3], and "War and Peace" (WP) dataset prepared by Karpathy et al. (2016)[4]. We use two probabilistic evaluation metrics that are standard in language modeling tasks: *Cross Entropy* and *Bits per Character (BpC)*. Depending on the dataset, we use one or the other such that we can directly compare to published results.

Tables 1 and 2 present results in terms of the context size (n) for the PTB and WP tests respectively. The column "UB" shows an upper-bound on the performance metric using a context of size n. This is computed directly using the expected counts on the test set to compute the conditional distribution. If we were able to estimate these expectations perfectly, we would achieve the

[3] 49 characters; 5017k / 393k / 442k characters in the train / dev / test portions.

[4] 84 symbols; 2658k / 300k / 300k characters in the train / dev / test portions.

n	UB	KN	Spectral longest	Spectral interp.	Spectral size $\mathbf{H}$
3	2.60	2.63	2.63	2.63	102
4	1.94	2.01	2.02	2.03	750
5	1.51	1.67	1.70	1.68	1,661
6	1.23	1.54	1.62	1.55	6,360
7	0.98	1.49	1.65	1.49	13,992
8	0.78	1.47	1.67	1.47	35,263
9	0.59	1.47	1.68	1.45	69,292
10	0.46	1.47	1.67	1.45	137,370

Table 1: Bits-per-character on the PTB test set.

n	UB	KN	FNN	ME	Spectral long.	Spectral int.	Spectral size $\mathbf{H}$
3	1.86	1.93	1.93	1.95	1.95	1.95	174
4	1.38	1.52	1.55	1.59	1.57	1.55	1,258
5	1.06	1.31	1.45	1.43	1.41	1.36	3,278
6	0.82	1.23	1.34	1.36	1.39	1.29	11,859
7	0.62	1.20	1.32	1.33	1.42	1.25	26,848
8	0.46	1.19	-	1.30	1.46	1.24	62,628
9	0.32	1.19	-	1.30	1.47	1.24	121,534
10	0.22	1.19	-	1.30	1.47	1.24	224,159

Table 2: Cross-entropy on the WP test set.

reported performance. As the two tables show, a context of size 10 already gives a high upper-bound, suggesting that we can achieve good performance using a fixed but large horizon.

The tables show results of the spectral language model for different context sizes, using expectations from the "longest" context or "interpolated" expectations. A clear trend is that the results improve with the context length, achieving a stable performance for $n = 10$. It is also clear that the interpolated predictions work much better than simply using the longest context. Table 2 also compares to a MaxEnt model (labeled "ME"), which is an interpolation model of Eq.3 but uses empirical expectations $f_{\mathcal{T}}(x)$ computed from training counts instead of those given by the spectral PNFA. Clearly, the expectations given by the PNFA generalize better and lead to improvements.

The last column of the two tables shows the number of rows (and columns) of the (square) Hankel matrix we factorize for each context size. This gives an idea of the cost of the estimation algorithm, which goes from a few seconds to a few hours, depending on the matrix size.[5] Following the theory behind Quattoni et al. (2017), this number is an upper bound on the size of the minimal PNFA that reproduces exactly the expected counts of training substrings.

The tables include a column "KN" with the results of an ngram language model estimated with Kneser-Ney interpolation (Kneser and Ney, 1995; Chen and Goodman, 1999). Looking at the results on the PTB data in Table 1, our interpolated model performs equally well, and sometimes better, than the KN models using the same context length. Mikolov et al. (2012) reports the performance of other models: a feed-forward neural network[6] obtains 1.57, which our model improves with contexts of $n = 6$ or larger; an RNN works at 1.41, slightly better than our best result of 1.45. Their best result is of 1.37 for a MaxEnt model with context length of $n = 14$ engineered for scalability.

For the WP test in Table 2, our model and the KN model perform similarly, with some slight improvements by the KN model. The table also includes the results of a feed-forward neural network (FNN) for increasing orders, by Karpathy et al. (2016). We observe that our interpolated model works better, with our best result at 1.24. They also report the results of an RNN obtaining 1.24, and of LSTM and GRU which both obtain 1.08.

5 Conclusions

In this paper we presented experiments using character-based spectral ngram language models. We combine two key ideas: a) modeling of long-range dependencies via the basis selection of long substring moments by Quattoni et al. (2017); and b) efficient optimization of arbitrary prediction losses (e.g. cross-entropy) via a loss refinement step. With these two ideas, we can improve the performance of spectral learning for PNFA, and bring the results of spectral models closer to the state-of-the-art.

The ability of the spectral method for PNFA to estimate substring expectations can be exploited in other contexts. For example, we are interested in word-level language models that make use of character-level PNFA to compute expectations, which is useful to make predictions on words and substrings which do not appear in training.

It is also interesting to consider a PNFA as a special case of an RNN which uses linear transi-

[5] Note that without the scalability trick, the Hankel matrices would be simply too big (in the order of millions of rows and columns) to practically run any experiment. It should be clear, though, that this is the contribution in Quattoni et al. (2017), not of this paper.

[6] However, they do not report the order of that model.

tions. Given that we obtain similar results than feed-forward NN and some RNN, this suggests that some forms of non-linearities can be approximated by linear models, with the advantage that some computations (mainly, expectations) can be done exactly.

Acknowledgments

We are grateful to Matthias Gallé for the discussions around this work, as well as to the anonymous reviewers for their useful feedback.

References

Raphaël Bailly, François Denis, and Liva Ralaivola. 2009. Grammatical inference as a principal component analysis problem. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 33–40, New York, NY, USA. ACM.

Borja Balle, Xavier Carreras, Franco M. Luque, and Ariadna Quattoni. 2014. Spectral Learning of Weighted Automata: A Forward-Backward Perspective. *Machine Learning*, 96(1):33–63.

Borja Balle, Ariadna Quattoni, and Xavier Carreras. 2011. A spectral learning algorithm for finite state transducers. In *Proceedings of the 2011th European Conference on Machine Learning and Knowledge Discovery in Databases - Volume Part I*, ECMLPKDD'11, pages 156–171, Berlin, Heidelberg. Springer-Verlag.

Borja Balle, Ariadna Quattoni, and Xavier Carreras. 2012. Local loss optimization in operator models: A new insight into spectral learning. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, ICML'12, pages 1819–1826, USA. Omnipress.

Stanley Chen. 2009. Performance prediction for exponential language models. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 450–458. Association for Computational Linguistics.

Stanley F. Chen and Joshua Goodman. 1999. An empirical study of smoothing techniques for language modeling. *Computer Speech & Language*, 13(4):359 – 394.

Shay B. Cohen, Karl Stratos, Michael Collins, Dean P. Foster, and Lyle Ungar. 2012. Spectral learning of latent-variable pcfgs. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 223–231, Jeju Island, Korea. Association for Computational Linguistics.

Daniel Hsu, Sham M Kakade, and Tong Zhang. 2012. A spectral algorithm for learning hidden markov models. *Journal of Computer and System Sciences*, 78(5):1460–1480.

Daniel J. Hsu, Sham M. Kakade, and Tong Zhang. 2009. A spectral algorithm for learning hidden markov models. In *COLT 2009 - The 22nd Conference on Learning Theory, Montreal, Quebec, Canada, June 18-21, 2009*.

Andrej Karpathy, Justin Johnson, and Fei-Fei Li. 2016. Visualizing and understanding recurrent networks. In *ICLR Workshop Track*.

Reinhard Kneser and Hermann Ney. 1995. Improved backing-off for m-gram language modeling. In *In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, volume I, pages 181–184.

Tomáš Mikolov, Ilya Sutskever, Anoop Deoras, Hai-Son Le, Stefan Kombrink, and Jan Cernocky. 2012. Subword language modeling with neural networks. *preprint (http://www. fit. vutbr. cz/imikolov/rnnlm/char. pdf)*.

Ankur P. Parikh, Avneesh Saluja, Chris Dyer, and Eric Xing. 2014. Language modeling with power low rank ensembles. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1487–1498, Doha, Qatar. Association for Computational Linguistics.

Ariadna Quattoni, Borja Balle, Xavier Carreras, and Amir Globerson. 2014. Spectral regularization for max-margin sequence tagging. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1710–1718. JMLR Workshop and Conference Proceedings.

Ariadna Quattoni, Xavier Carreras, and Matthias Gallé. 2017. A Maximum Matching Algorithm for Basis Selection in Spectral Learning. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pages 1477–1485, Fort Lauderdale, FL, USA. PMLR.

Roni Rosenfeld. 1994. *Adaptive statistical language modeling: A maximum entropy approach*. Ph.D. thesis, Carnegie Mellon University.