NONUNIFORM MARKOV MODELS

Eric Sven Ristad *Robert G. Thomas*

Department of Computer Science
Princeton University
Princeton, NJ 08544-2087
{ristad,rgt}@cs.princeton.edu

ABSTRACT

We propose a new way to model conditional independence in Markov models. The central feature of our nonuniform Markov model is that it makes predictions of varying lengths using contexts of varying lengths. Experiments on the Wall Street Journal reveal that the nonuniform model performs slightly better than the classic interpolated Markov model of Jelinek and Mercer (1980). This result is somewhat remarkable because both models contain identical numbers of parameters whose values are estimated in a similar manner. The only difference between the two models is how they combine the statistics of longer and shorter strings.

1. INTRODUCTION

A statistical language model assigns probability to strings of arbitrary length. Unfortunately, it is not possible to gather reliable statistics on strings of arbitrary length from a finite corpus. Therefore, a statistical language model must decide that each symbol in a string depends on at most a small, finite number of other symbols in the string. In this report, we propose a new way to model conditional independence in Markov models. The central feature of our nonuniform Markov model is that it makes predictions of varying lengths using contexts of varying lengths.

We believe that our work has two contributions to offer to the field of Markov modeling. The first contribution is our interpretation of the interpolation parameters as beliefs about conditional independence. Prior work on interpolated Markov models has interpreted the interpolation parameters as smoothing the "specific probabilities" with the "general probabilities" [6, 8]. Our interpretation gives rise to the second contribution of our work, namely, a class of *nonuniform* Markov models that make predictions of varying lengths using contexts of varying lengths. Nonuniform predictions is a principled way to perform alphabet extension, that is, to make a string become a symbol in the alphabet, an *ad hoc* technique that can improve model performance [4].

The remainder of this article consists of four sections. Section 2. considers two generative interpretations of the interpolated Markov model: the context model and our *nonuniform model*. Section 3. provides a dynamic programming algorithm to evaluate the probability of a string according to a nonuniform model, while section 4. provides an algorithm to optimize the parameters of a nonuniform model on a training corpus. Section 5. compares the performance of the classic interpolated Markov model and the nonuniform model on the Wall Street Journal. The nonuniform model performs slightly better than the classic model under equivalent experimental conditions. This result is somewhat remarkable, since the only difference between these two models is how they interpret the interpolation parameters.

Let us first define our notation. Let A be a finite alphabet of distinct symbols, $|A| = k$, and let $x^T \in A^T$ denote an arbitrary string of length T over the alphabet A. Then x_i^j denotes the substring of x^T that begins at position i and ends at position j. We abbreviate the unit length substring x_i^i as x_i and the length t prefix of x^T as x^t.

2. TWO INTERPOLATED MODELS

Recall that an interpolated Markov model $\phi = \langle n, A, \delta, \lambda \rangle$ consists of a maximal string length n, a finite alphabet A, a set of string probabilities $\delta : A^{\leq n} \to [0,1]$, and the state-conditional interpolation parameters $\lambda : A^{<n} \to [0,1]$. Given a string y^l, $l \leq n$, the string probabilities $\delta(y^l)$ are typically their empirical probabilities in a training corpus.

We now consider two generative interpretations of the interpolated Markov model: the classic context model and our nonuniform model. The only difference between these two models will be how the interpolation parameters λ are interpreted. In each case, we let $\bar{p}_c(i|x_{t-m+1}^t)$ be the probability that we pick a context of length i in the history x_{t-m+1}^t and let $\bar{p}_v(y_1^j|x_{t-i+1}^t)$ be the probability that we make a prediction y_1^j of length j in the chosen context x_{t-i+1}^t.

2.1. Context Model

In the interpolated context model, the interpolation parameters are understood as smoothing the conditional probabilities estimated from longer histories with those estimated from shorter histories [6, 8]. Longer histories support stronger predictions, while shorter histories have more accurate statistics. Interpolating the predictions from histories of different lengths results in more accurate predictions than can be obtained from any fixed history length. This interpretation of the interpolation parameters was originally proposed by Jelinek and Mercer [6]. It leads to the following generation algorithm, where the hidden transition from a longer context to a shorter context (line 3) is temporary, used only for the current prediction (line 4).

CONTEXT-GENERATE(T,ϕ)
1. Initialize $t := 0$; $x_1^0 := \epsilon$;
2. Until $t = T$
3. Pick context length i in $[0, \min(t, n-1)]$
$\bar{p}_c(i|x^t) = \lambda(x_{t-i+1}^t) \prod_{l=\min(t,n-1)}^{i+1} (1 - \lambda(x_{t-l+1}^t))$
4. Make one symbol prediction y^1
$\bar{p}_v(y^1|x_{t-i+1}^t) = \delta(y^1|x_{t-i+1}^t, i+1)$
5. Extend history x_1^t by prediction y^1
$x_1^{t+1} := x_1^t y^1$; $t := t+1$;
6. return(x^T);

The probability $p_c(x_i|x^{i-1}, \phi)$ assigned by an interpolated context model ϕ to a symbol x_i in the history x^{i-1} has a simple iterative form (1),

$$\begin{aligned} p_c(x_i|x^{i-1}, \phi) &= \lambda(x^{i-1})\delta(x_i|x^{i-1}) \\ &\quad +(1 - \lambda(x^{i-1}))p_c(x_i|x_2^{i-1}, \phi) \end{aligned} \tag{1}$$

2.2. Nonuniform Model

We propose to interpret the interpolation parameter $\lambda(x^i)$ as our degree of belief that the next $n-i$ symbols depend on x_1. Our interpretation has two implications. The first implication, as in the uniform model, is that we should transition from a context x^i to its proper suffix x_2^i with probability $1-\lambda(x^i)$. This expresses our belief of degree $1-\lambda(x^i)$ that the future does not depend on x_1. The second implication, which is unique to the nonuniform model, is that we should transition from a shorter prediction y^{j-1} to a longer prediction y^j in the chosen context x^i with probability $\lambda(x^i y^{j-1})$. This implication follows from our belief of degree $\lambda(x^i y^{j-1})$ that the future depends on the entire string $x^i y^{j-1}$ and does not depend on any symbol further in the past. Our novel interpretation leads to the following *nonuniform* generation algorithm.

NONUNIFORM-GENERATE(T,ϕ)
1. Initialize $t := 0$; $x_1^0 := \epsilon$;
2. Until $t = T$
3. Pick context length i in $[0, \min(t, n-1)]$
$\bar{p}_c(i|x^t) = \lambda(x_{t-i+1}^t) \prod_{l=\min(t,n-1)}^{i+1} (1 - \lambda(x_{t-l+1}^t))$
4. $c := x_{t-i+1}^t$; $j_{\max} := \max(n-i, T-t)$;
5. Pick prediction y_1^j of length j in $[1, j_{\max}]$
$\bar{p}_v(y_1^j|c) = (1 - \lambda(cy_1^j))\delta(y_j|cy_1^{j-1}, i+j)$
$\prod_{l=1}^{i-1} \lambda(cy_1^l)\delta(y_l|cy_1^{l-1}, l+i)$
where $\lambda(cy_1^{j_{\max}}) \doteq 0$.
6. Extend history x_1^t by prediction y_1^j
$x_1^{t+j} := x_1^t y_1^j$; $t := t+j$;
7. return(x^T);

3. EVALUATION

The following dynamic programming algorithm evaluates the probability of a string x^T according to a nonuniform model ϕ in $O(n^2T)$ time and $O(T)$ space. The resource requirements of the algorithm may be reduced to $O(nT)$ time and $O(n)$ space at a significant expense in clarity [11]. Note that $\lambda(x_{t-i+1}^{t+j_{\max}}) = 0$ for $j_{\max} = \min(T-t, n-i)$.

NONUNIFORM-EVALUATE(x^T,ϕ)
1. For $t = 2$ to T [$\alpha_t := 0$]; $\alpha_1 := 1$;
2. For $t = 1$ to $T-1$
3. $p_c = 1$;
4. for $i = \min(t, n-1)$ to 0
5. $\bar{p}_c := \lambda(x_{t-i+1}^t)p_c$; $p_v := 1$;
6. for $j = 1$ to $\min(T-t, n-i)$
7. $\bar{p}_v := (1 - \lambda(x_{t-i+1}^{t+j}))\delta(x_{t+1}^{t+j}|x_{t-i+1}^t, i+j)p_v$;
8. $\alpha_{t+j} := \alpha_{t+j} + \alpha_t \bar{p}_c \bar{p}_v$;
9. $p_v := \lambda(x_{t-i+1}^{t+j})p_v$;
10. $p_c := (1 - \lambda(x_{t-i+1}^t))$;
11. return(α_T);

The α_t variable stores the total probability $p(x^t|\phi, t)$ for the substring x^t.

4. DELETED ESTIMATION

In this section, we formulate an expectation maximization (EM) algorithm for the nonuniform Markov model. Our development follows the traditional lines established for the hidden Markov model [2, 3]. We begin by defining our forward and backward variables. The forward variable $\alpha_t(i,j)$ contains the probability of generating the first t symbols of the history, picking a context of length i and then making a prediction of length j, according to the model ϕ.

$$\alpha_t(i,j) \doteq p(h = x_1^t, c = x_{t-i+1}^t, v = x_{t+1}^{t+j}|\phi, T) \tag{2}$$

The following algorithm calculates all $\alpha_t(i,j)$ values in $O(n^2T)$ time and $O(n^2T)$ space.

FORWARD(x^T,ϕ)
1. For $j = 1$ to n [$\alpha_0(0,j) := \bar{p}_v(x_1^j|\epsilon)$;];
2. For $t = 1$ to T
3. $\alpha_t := \sum_{j=1}^{\min(t,n)} \sum_{i=0}^{\min(n-j,t-j)} \alpha_{t-j}(i,j)$;
4. For $i = 0$ to $\min(t, n-1)$
5. For $j = 1$ to $\min(T-t, n-i)$
6. $\alpha_t(i,j) := \alpha_t \bar{p}_c(i|x_1^t)\bar{p}_v(x_{t+1}^{t+j}|x_{t-i+1}^t)$;
7. return(α);

The backward variable $\beta_t(i,j)$ contains the probability of generating the final $T-t$ symbols in the string x_1^T, given that the history is x_1^t and that we have chosen to make a prediction of length j in a context of length i according to the model ϕ.

$$\begin{aligned} \beta_t(i,j) &\doteq p(x_{t+1}^T|h = x_1^t, c = x_{t-i+1}^t, v = x_{t+1}^{t+j}|\phi, T) \\ &= p(x_{t+j+1}^T|x_1^{t+j}, \phi) = \beta_{t+j} \end{aligned} \tag{3}$$

The following algorithm calculates all β_t values in $O(n^2T)$ time and $O(T)$ space. Note that we need only maintain a one dimensional table of β values because $\beta_t(i,j) = \beta_{t+j}$ for all i, j.

BACKWARD(x^T,ϕ)
1. $\beta_T := 1$;
2. For $t = T-1$ to 0
3. $\beta_t := 0$;
4. For $i = 0$ to $\min(t, n-1)$
5. For $j = 1$ to $\min(T-t, n-i)$
6. $\beta_t \mathrel{+}= \bar{p}_c(i|x_1^t)\bar{p}_v(x_{t+1}^{t+j}|c = x_{t-i+1}^t)\beta_{t+j}$;
7. return(β);

The forward and backward variables allow us to efficiently calculate the posteriori probability of every hidden transition in our model, as represented by the following $\gamma_t(i,j)$ variable.

$$\begin{aligned}\gamma_t(i,j) &\doteq p(c = x_{t-i+1}^{t}, v = x_{t+1}^{t+j} | x_1^T, \phi) \\ &= \alpha_t(i,j)\beta_t(i,j)/p(x_1^T|\phi)\end{aligned} \quad (4)$$

We sum the γ values to obtain the expected number of times that the nonuniform model transitioned from a longer context to a shorter one, or from a shorter prediction to a longer one. We use two variables to keep track of our expectations: $\lambda^+(y^l)$ accumulates the number of times that we used y^l to condition our prediction when it was possible to do so, while $\lambda^-(y^l)$ accumulates the number of times that we could have used y^l to condition our prediction but chose a proper suffix instead. The following algorithm accumulates all $\lambda^+(y^l)$ and $\lambda^-(y^l)$ values in $O(n^3T)$ time and $O(n^2T)$ space. The full paper [11] presents an alternate expectation step algorithm that accumulates all expectations in $O(nT)$ time and space.

```
EXPECTATION-STEP(x^T,φ,λ+,λ-)
1.  α = FORWARD(x^T,φ);
2.  β = BACKWARD(x^T,φ);
3.  For t = 1 to T − 1
4.    For i = 0 to min(t, n − 1)
5.      For j = 1 to min(T − t, n − i)
6.        λ+(x_{t−i+1}^{t}) += γ_t(i,j);
7.        λ-(x_{t−i+1}^{t+j}) += γ_t(i,j);
8.        For l = i + 1 to min(t, n − 1)
9.          λ-(x_{t−l+1}^{t}) += γ_t(i,j);
10.       For l = j + 1 to min(T − t, n − i)
11.         λ+(x_{t−i+1}^{t+l}) += γ_t(i,j);
```

Having done all the work in the expectation step, the maximization step is straightforward.

```
MAXIMIZATION-STEP(φ,λ+,λ-)
1.  For all strings y^l in A^{<n}
2.    λ(y^l) := λ+(y^l)/(λ+(y^l) + λ-(y^l));
```

The following DELETED-ESTIMATION() algorithm estimates the parameters of an interpolated model ϕ using a set $\mathbf{B}$ of blocks of text. For each iteration, we delete one block B_i from the set $\mathbf{B}$, initialize the string probabilities δ to their empirical probabilities in the remaining blocks $\mathbf{B} - B_i$ (line 4), and then perform an expectation step on the deleted block B_i (line 5). After all blocks have been deleted, we update our model parameters (line 6).

```
DELETED-ESTIMATION(B,φ)
1.  Until convergence
2.    Initialize λ+, λ- to zero;
3.    For each block B_i in B
4.      Initialize δ using B − B_i;
5.      EXPECTATION-STEP(B_i,φ,λ+,λ-);
6.    MAXIMIZATION-STEP(φ,λ+,λ-);
7.  Initialize δ using B;
```

5. EXPERIMENTAL RESULTS

In this section we compare the performance of the interpolated context model and the nonuniform model on the Wall Street Journal. (Recall that the interpolated context model is the classic interpolated Markov model of Jelinek and Mercer [6].) We performed two sets of experiments. The first set of experiments was with the 6.2 million word WSJ 1989 corpus. The goal of these initial experiments was to better understand how initial parameter values affect model performance. The second set of experiments was with the 42.3 million word WSJ 1987-89 corpus. In order to assess the possible value of our language models to speech recognition, we used verbalized punctuation and a vocabulary of approximately 20,000 words chosen from both training and test sets. All out-of-vocabulary words were mapped to a single unique OOV symbol. In all experiments we used 90% of the corpus for training and 10% for testing. No parameter tying or parameter selection was performed. We report performance as test message perplexity.

We set the δ parameters to be the empirical probabilities in the training data and then optimized the λ parameters on the training data using deleted estimation [6, 1]. We report the best numbers for each model, as though an oracle told us when to stop running deleted estimation. We considered three initial estimates for the λ parameters: the uniform estimate 0.5, the Jeffreys-Perks rule of succession [5, 9, 7], and the natural law of succession [10]. Jeffreys-Perks assigns relatively low probability to $\lambda(x^*)$, while the natural law assigns relatively high probability to $\lambda(x^*)$. The best performance for higher model orders was achieved with uniform initialization in all of our experiments, both before and after optimization via deleted estimation. Regardless of how the λ parameters were initialized, the nonuniform model performed slightly better than the classic interpolated context model under equivalent experimental conditions.

5.1. WSJ 1989

The first set of experiments was on the 1989 Wall Street Journal corpus, which contains 6,219,350 words. Our vocabulary consisted of the 20,293 words that occurred at least 10 times in the entire WSJ 1989 corpus. The goal of these initial experiments was to better understand how initial values affect model performance.

5.1.1. Before Optimization

The following table reports test message perplexities for WSJ 1989 before the λ parameters were optimized using deleted interpolation. The best results for both models are obtained when the λ parameters are initialized uniformly. Before optimization the interpolated context model performs better than the nonuniform model.

	Context Model		
N	Jeffrey-Perks	Natural Law	0.5
2	284.9	**188.2**	215.9
3	248.1	148.7	**136.0**
4	241.6	155.0	**130.0**
5	239.6	161.7	**131.3**
6	238.7	165.7	**132.6**

N	Nonuniform Model		
	Jeffrey-Perks	Natural Law	0.5
2	276.8	197.6	209.6
3	235.8	175.4	138.4
4	229.3	196.3	138.3
5	227.6	211.4	142.6
6	226.9	219.4	145.2

5.1.2. *After Optimization*

The following table reports test message perplexities for WSJ 1989 after optimization via deleted estimation. All models were trained using deleted estimation with 22 blocks on the first 90% of the corpus and then tested on the remaining 10% of the corpus. The nonuniform model slightly outperforms the context model for $n > 3$. The best results for both models are obtained when the λ parameters are initialized uniformly. The nonuniform model is less sensitive to the initial λ estimates than the context model.

N	Context Model		
	Jeffrey-Perks	Natural Law	0.5
2	175.3	**175.2**	**175.2**
3	122.1	121.8	**121.2**
4	115.8	115.9	114.0
5	114.5	115.4	112.6
6	114.1	115.6	112.3

N	Nonuniform Model		
	Jeffrey-Perks	Natural Law	0.5
2	177.7	177.6	177.7
3	121.6	121.6	**121.2**
4	113.6	114.1	**113.2**
5	111.9	113.0	**111.4**
6	111.5	112.9	**111.0**

5.2. WSJ 1987-89

The second set of experiments was on the 1987-89 Wall Street Journal corpus, which contains 42,373,513 words. Our vocabulary consisted of the 20,092 words that occurred at least 63 times in the entire WSJ 1987-89 corpus. The goal of these experiments was to produce competitive results for the context model, in order to compare those results to those achieved by the nonuniform model.

5.2.1. *Before Optimization*

The following table reports test message perplexities for WSJ 1987-89 before optimization via deleted estimation. All λ values were initialized uniformly.

N	Context Model	Nonuniform Model
2	198.2	**190.1**
3	107.5	**106.1**
4	**97.7**	100.4

5.2.2. *After Optimization*

The following table reports test message perplexities for WSJ 1987-89 after optimization via deleted estimation. All λ values were initialized uniformly, trained using deleted estimation with 152 blocks on the first 90% of the corpus, and then tested on the remaining 10% of the corpus. The nonuniform model performs slightly better than the context model for $n > 2$.

N	Context Model	Nonuniform Model
2	**150.7**	151.7
3	93.4	**93.3**
4	85.7	**84.4**

6. CONCLUSION

We have proposed a nonuniform Markov model, that makes predictions of varying lengths using contexts of varying lengths, and demonstrated that the nonuniform model slightly outperforms the interpolated context model on natural language text. This result is somewhat remarkable when we consider that both models are based on the statistics of fixed-length strings, and that both models contain identical numbers of parameters whose values are estimated using deleted estimation. The only difference between the two models is how they combine the statistics of longer and shorter strings.

REFERENCES

[1] Bahl, L. R., Brown, P. F., de Souza, P. V., Mercer, R. L., and Nahamoo, D. A fast algorithm for deleted interpolation. In *Proc. EUROSPEECH '91* (Genoa, 1991), pp. 1209–1212.

[2] Baum, L., and Eagon, J. An inequality with applications to statistical estimation for probabilistic functions of a Markov process and to models for ecology. *Bull. AMS 73* (1967), 360–363.

[3] Baum, L., Petrie, T., Soules, G., and Weiss, N. A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *Ann. Math. Stat. 41* (1970), 164–171.

[4] Jeanrenaud, P., Eide, E., Chaudhari, U., McDonough, J., Ng, K., Siu, M., and Gish, H. Reducing word error rate on conversational speech from the Switchboard corpus. In *ICASSP 95* (1995), pp. 53–56.

[5] Jeffreys, H. An invariant form for the prior probability in estimation problems. *Proc. Roy. Soc. (London) A 186* (1946), 453–461.

[6] Jelinek, F., and Mercer, R. L. Interpolated estimation of Markov source parameters from sparse data. In *Pattern Recognition in Practice* (Amsterdam, May 21–23 1980), E. S. Gelsema and L. N. Kanal, Eds., North Holland, pp. 381–397.

[7] Krichevskii, R. E., and Trofimov, V. K. The performance of universal coding. *IEEE Trans. Information Theory IT-27*, 2 (1981), 199–207.

[8] MacKay, D. J., and Peto, L. C. B. A hierarchical Dirichlet language model. *Natural Language Engineering 1*, 1 (1994).

[9] Perks, W. Some observations on inverse probability, including a new indifference rule. *J. Inst. Actuar. 73* (1947), 285–312.

[10] Ristad, E. S. A natural law of succession. Tech. Rep. 495-95, Department of Computer Science, Princeton University, Princeton, NJ, May 1995.

[11] Ristad, E. S., and Thomas, R. G. Nonuniform Markov models. Tech. Rep. 536-96, Department of Computer Science, Princeton University, Princeton, NJ, November 1996.