# Principles of Non-stationary Hidden Markov Model and Its Applications to Sequence Labeling Task

Xiao JingHui, Liu BingQuan, and Wang XiaoLong

School of Computer Science and Techniques, Harbin Institute of Technology, Harbin, 150001, China {xiaojinghui, liubq, wangxl}@insun.hit.edu.cn

Abstract. Hidden Markov Model (Hmm) is one of the most popular language models. To improve its predictive power, one of Hmm hypotheses, named limited history hypothesis, is usually relaxed. Then Higher-order Hmm is built up. But there are several severe problems hampering the applications of highorder Hmm, such as the problem of parameter space explosion, data sparseness problem and system resource exhaustion problem. From another point of view, this paper relaxes the other Hmm hypothesis, named *stationary (time invariant)* hypothesis, makes use of time information and proposes a non-stationary Hmm (NSHmm). This paper describes NSHmm in detail, including its definition, the representation of time information, the algorithms and the parameter space and so on. Moreover, to further reduce the parameter space for mobile applications, this paper proposes a variant form of NSHmm (VNSHmm). Then NSHmm and VNSHmm are applied to two sequence labeling tasks: pos tagging and pinyin-tocharacter conversion. Experiment results show that compared with Hmm, NSHmm and VNSHmm can greatly reduce the error rate in both of the two tasks, which proves that they have much more predictive power than Hmm does.

# 1 Introduction

Statistical language model plays an important role in natural language processing and great efforts are devoted to the research of language modeling. Hidden Markov Model (Hmm) is one of the most popular language models. It was first proposed by IBM in speech recognition [1] and achieved great success. Then Hmm has a wide range of applications in many domains, such as OCR [2], handwriting recognition [3], machine translation [4], Chinese pinyin-to-character conversion [5] and so on.

To improve Hmm's predictive power, one of Hmm hypotheses [6] named *limited history hypothesis*, is usually relaxed and higher-order Hmm is proposed. But as the order of Hmm increases, its parameter space explodes at an exponential rate, which may result in several severe problems, such as data sparseness problem [7], system resource exhaustion problem and so on. From another point of view, this paper relaxes the other Hmm hypothesis, named *stationary hypothesis*, makes use of time information and proposes non-stationary Hmm (NSHmm). This paper first defines NSHmm in a formalized form, and then discusses how to represent time information in NSHmm. After that, the algorithms of NSHmm are provided and the parameter space complexity is calculated. Moreover, to further reduce the parameter space, a

variant form of NSHmm (VNSHmm) is proposed later. At last, NSHmm and VNSHmm are applied to two sequence labeling tasks: pos tagging and pinyin-tocharacter conversion. As the experiment results show, compared with Hmm, NSHmm and VNSHmm can greatly reduce the error rate in the both two tasks.

The rest of this paper is structured as follows: in section 2 we briefly review the definition of standard Hmm. In section 3, NSHmm is proposed and the relative questions are discussed in detail. Experiments and results are discussed in section 4. Finally, we give our conclusions in section 5 and plan the further work in section 6.

# 2 Hidden Markov Model

Hmm is a function of Markov process and can be mathematically defined as a fivetuple M =  $\langle \Omega, \Sigma, \rho, \alpha, \theta \rangle$  which consists of:

- 1. A finite set of (hidden) states  $\Omega$ .
- 2. A finite set of (observed) symbols  $\Sigma$ .
- 3. A state transition function  $\rho: \Omega \times \Omega \rightarrow [0, 1]$ .
- 4. A symbol emission function  $\alpha$ :  $\Omega \times \Sigma \rightarrow [0, 1]$ .
- 5. And an initial state probability function  $\theta$ : Omega -> [0, 1].

The functions of  $\rho$ ,  $\alpha$  and  $\theta$  are usually estimated by MLE principle on large scale corpus. Based on the above definition, Hmm makes two hypotheses at the same time:

- 1. *Limited history hypothesis*: the current state is completely decided by the last state before, but irrelative to the entire state history.
- 2. *Stationary hypothesis*: the state transition function  $\rho$  is completely determined by states, but irrelative to the time when state transition occurs. So it is with the symbol emission function.

There are three fundamental questions and a series of corresponding algorithms for Hmm:

- 1. Given Hmm, how to calculate the probability of a sequence observation? Forward algorithm and backward algorithm can handle that question.
- 2. Given Hmm and an observation sequence, how to find the best state sequence to explain the observation? Viterbi algorithm can fulfill that task.
- 3. Given an observation sequence, how to estimate the parameters of Hmm to best explain the observed data? Baum-Welch algorithm can solve that problem.

Hmm is a popular language model and has been applied to many tasks in natural language processing. For example, in pos tagging, the word sequence is taken as the observation of Hmm, and the pos sequence as the hidden state chain. Viterbi algorithm can find the best pos sequence corresponding to the word sequence.

# 3 Non-stationary Hidden Markov Model

### 3.1 Motivation

There are many approaches to improve the predictive power of Hmm in practice. For example, factorial Hmm [8] is proposed by decomposing the hidden state

representation into multiple independent Markov chains. In speech recognition, a factorial Hmm can represent the combination of multiple signals which are produced independently and the characteristics of each signal are described by a distinct Markov chain. And some Hmms use neural networks to estimate phonetic posterior probability in speech recognition [9]. The input layer of the network typically covers both the past states and the further states. However, from the essential definition of Hmm, there are two ways to improve the predictive power of Hmm. One approach is to relax the limited history hypothesis and involve more history information into language model. The other is to relax the stationary hypothesis and make use of time information. In recent years, much research focuses on the first approach [10] and higher-order Hmm is built up. But as the order increases, the parameter space explodes at such an exponential rate that training corpus becomes too sparse and system resource exhausts soon. This paper adopts the second approach and tries to make good use of time information. Then NSHmm is proposed. Since there is no theoretical conflict between NSHmm and high-order Hmm, the two models can be combined together in proper conditions.

#### 3.2 Definition for NSHmm

Similarly with Hmm, NSHmm is also mathematically defined as a five-tuple  $M = \langle \Omega, \Sigma, \rho', \alpha', \theta' \rangle$  which consists of:

- 1. A finite set of (hidden) states  $\Omega$ .
- 2. A finite set of (observed) symbols  $\Sigma$ .
- 3. A state transition function  $\rho : \Omega \times \Omega \times t \rightarrow [0, 1]$ .
- 4. A symbol emission function  $\alpha : \Omega \times \Sigma \times t \rightarrow [0, 1]$ .
- 5. And an initial state probability function  $\theta': \Omega \times t \rightarrow [0, 1]$ .

In the above definition, *t* is the time variable indicating when state transition or symbol emission occurs. Different from Hmm's definition,  $\rho'$ ,  $\alpha'$  and  $\theta'$  are all the functions of *t*. And they can still be estimated by MLE principle on large scale corpus. This key question of NSHmm is how to represent time information. We'll discuss that question in the next section.

### 3.3 Representation of Time Information

Since time information is to describe when the events of Hmm (e.g. state transition or symbol emission) occur, a natural way is to use the event index in Markov chain to represent the time information. But there are two serious problems with that method. Firstly, index has different meanings in the Markov chains of different length. Secondly, since a Markov chain may have arbitrary length, the event index can be any natural number. However, computer system can only deal with finite value. A refined method is to use the ratio of the event index and the length of Markov chain which is a real number of the range [0, 1]. But there are infinite real numbers in the range [0, 1]. In this paper, we divide the range [0, 1] into several equivalence classes (bins) and each class share the same time information. When training NSHmm, the functions of  $\rho'$ ,  $\alpha'$  and  $\theta'$  should be estimated in each bin respectively according to their time information. And when they are accessed, they should also get the value in the

according bin. For example, the state transition function  $\rho^{'}$  can be estimated by the formula below:

$$p_{ijt} = \frac{C(i, j, t)}{C(i, t)} \tag{1}$$

where C(i, j, t) is the co-occurrence frequency of state i and state j at time t and it can be estimated by counting the co-occurrence times of state i and state j in the  $t^{\text{th}}$  bin in each sentence of corpus. C(i, t) is the frequency of state i at time t and can be estimated by counting the occurrence times of state i in the  $t^{\text{th}}$  bin in the sentence of corpus. And the result  $P_{ijt}$  is the transition probability between state i and j at time t. It's similar to estimate the functions of  $\alpha'$  and  $\theta'$ .

### 3.4 Algorithms on Non-stationary Hidden Markov Model

The three fundamental questions of Hmm also exist in NSHmm. The corresponding algorithms, such as forward algorithm, viterbi algorithm and Baum-Welch algorithm, can work well in NSHmm, except that they have to first calculate the time information and then compute the function values of  $\rho'$ ,  $\alpha'$  and  $\theta'$  according to the statistical information in the corresponding bins.

### 3.5 Space Complexity Analysis

In this section, we will analyze the space complexity of NSHmm. Compared with Hmm, some conclusions can be drawn at the end of this section. For simplicity and convenience, we define some notations below:

- The hidden state number n
- The observed symbol number m
- The bin number for NSHmm k

In Hmm and NSHmm, all system parameters are devoted to simulate the three functions of  $\rho$ ,  $\alpha$  and  $\theta$ . For Hmm, a vector of size *n* is usually used to store the initial probability of each state. An  $n \times n$  matrix is adopted to store the transition probabilities between every two states, and  $n \times m$  matrix to record the emission probabilities between states and observed symbols. The space complexity for Hmm is the sum of these three parts which is  $O(n+n\times n+n\times m)$ . For NSHmm, since  $\rho$ ,  $\alpha$  and  $\theta$  are all the functions of time *t*, time information should be counted in. An  $n \times n \times k$  matrix is used to store the transition probability between each state at different time and  $n \times m \times k$  matrix to keep the emission probability. Thus, the space complexity of NSHmm is  $O((n+n\times n+n\times m)\times k)$  which is *k* times than that of Hmm. As the analysis shows, the space complexity of NSHmm increases at a linear speed with *k*, rather than at an exponential speed as high-order Hmm dose. Moreover, as *k* is usually far below than *n*, NSHmm is much easier to avoid the problem of parameter space explosion.

#### 3.6 Variant Form of NSHmm

In this section, this paper proposes a variant form of NSHmm (VNSHmm). It's based on these facts: for some applications, such as on mobile platform, there is not enough system resource to build up a whole NSHmm. Then NSHmm has to be compressed. This paper constructs some statistical variables for time information and uses these statistical variables to substitute concrete time information in NSHmm. When computing the probability in VNSHmm, these statistical variables are combined together to calculate a coefficient for normal probability of Hmm.

Two statistical variables, expectation and variance of time information, are adopted in VNSHmm. And such assumptions are made that more weight should be awarded if the time of event occurring fits better with the training corpus, and less weight vice versa. The probability function in VNSHmm is defined as below:

$$p_t = \frac{1}{Z} e^{\alpha \times V / ((t-E)^2 + \beta)} \times p$$
<sup>(2)</sup>

where Z is a normalizing factor, and is defined as:

$$Z = \sum_{t=1}^{t=k} e^{\alpha \times V/((t-E)^2 + \beta)} \times p \tag{3}$$

The notations in the formulation (2) and (3) are described in the following:

- Current time information t
- Expectation of time information E
- Variance of time information V
- State transition probability ( or symbol emission probability ) p
- Adjusted coefficients  $\alpha$  and  $\beta$

 $p_t$  is descendent with the term *t*-*E* which defines the difference between current time and time expectation in training corpus. As the value of *t*-*E* decreases, *t* fits for training corpus better and more weight is added to  $p_t$ . For example, we take a Chinese sentence as a state chain of Markov process. The word "首先"(first of all) usually leads a sentence in training corpus. For test corpus, more weight should be given to  $p_t$  if 首先(first of all) appears at the beginning of the sentence, whereas less weight if at the sentence end.  $p_t$  is ascendant with the variance *V*. The item *V* is mainly used to balance the value of term *t*-*E* for some active states. For example, in Chinese, some adjectives, such as "美丽"(beautiful), can appear at any position of the sentence. Then it's unreasonable to decrease  $p_t$  just because the term *t*-*E* increases. In such a situation, the value of item *V* for "美丽"(beautiful) is usually bigger than that of those inactive states (e.g. 首先(first of all)). Then the item *V* can provide a balance for the value of *t*-*E*.

Since VNSHmm just makes use of expectation and variance, rather than the whole time information, its space complexity is equal to that of the NSHmm with only two bins, which is  $O((n+n\times n+n\times m)\times 2)$ .

# **4** Experiments

In the experiments, NSHmm and VNSHmm have been applied to two sequence labeling tasks: pos tagging and pinyin-to-character conversion. This paper will describe them in detail in the following two sections.

### 4.1 Pos Tagging

For pos tagging, this paper chooses the People's Daily corpus in 1998 which has been labeled by Peking University [11]. The first 5 month corpus is taken as training corpus and the  $6^{th}$  month as test corpus. Since most of pos-taggers are based on 2-order Hmm (trigram), 2-order NSHmm and 2-order VNSHmm are constructed respectively in the experiments.

We first calculate KL distances between the emission probability distribution of Hmm and the distributions of NSHmm at different time. Only when the distances are great, could NSHmm be expected to outperform Hmm; otherwise NSHmm would have similar performance as Hmm has. Since there are totally k different distance values for NSHmm with k bins, we just calculate the average distance for each NSHmm. The results are presented in table 1 as below:

 Table 1. Average KL Distances between Emission Probability Distributions of NSHmm and Hmm

Bin Number	K=1	K=2	K=3	K=4	K=5	K=6	K=7	K=8
Aver KL Dis	0	0.08	0.12	0.15	0.17	0.19	0.21	0.22

From table 1 we can see that as the bin number increases, the average KL distance become bigger and bigger, which indicates there is more and more difference between the emission probability distributions of Hmm and that of NSHmm. Similar results can be gotten by comparing state-transition-probability distributions of the two models. And as time information increases, we expect more predictive power for NSHmm.

To prove the effectiveness of NSHmm and VNSHmm, in the rest of this section, two sets of experiments, close test and open test, are performed. The results of close test are showed in table 2, figure 1 and the results of open test are presented in table 3, figure 2 as below.

Table 2. Pos Tagging Close Test

Bin Nu	mber	K=1	K=2	K=3	K=4	K=5	K=6	K=7	K=8
Hmm (ba	aseline)	6.04%							
NSHmm	Error Rate	6.04%	5.63%	5.55%	5.52%	5.47%	5.44%	5.42%	5.47%
	Reduction		6.79%	8.11%	8.61%	9.44%	9.93%	10.26%	9.43%
VNSHmm	Error Rate	6.04%	5.85%	5.85%	5.85%	5.85%	5.85%	5.85%	5.85%
	Reduction		3.15%	3.15%	3.15%	3.15%	3.15%	3.15%	3.15%



Fig. 1. Pos Tagging Close Test

Table 3. Pos Tagging Open Test

Bin Nu	ımber	K=1	K=2	K=3	K=4	K=5	K=6	K=7	K=8
Hmm (b	aseline)	6.99%							
NSHmm	Error Rate	6.99%	6.44%	6.39%	6.42%	6.40%	6.43%	6.47%	6.58%
	Reduction		7.87%	8.58%	8.15%	8.44%	8.01%	7.44%	5.87%
VNSHmm	Error Rate	6.99%	6.59%	6.59%	6.59%	6.59%	6.59%	6.59%	6.59%
	Reduction		5.72%	5.72%	5.72%	5.72%	5.72%	5.72%	5.72%



Fig. 2. Pos Tagging Open Test

As table 2 and table 3 have showed, no matter in close test or in open test, NSHmm and VNSHmm achieve much lower error rates than Hmm. NSHmm gets at most 10.26% error rate reduction and VNSHmm obtains 3.15% reduction in close test; and

they achieve 8.58% and 5.72% reductions respectively in open test. These facts prove that NSHmm and VNSHmm have much more predictive power than Hmm has. From figure 1 we can see that in close test, as the bin number increases, the error rate of NSHmm is decreased constantly, which proves that the improvement of NSHmm is due to the increasing time information. But in the open test as figure 2 shows, the error rate stops decreasing after k=3. That is because of the overfitting problem. As a consequence, this paper suggests k=3 in NSHmm for pos tagging task. From figure 1 and figure 2, VNSHmm performs stably after k=2, which indicates a small number of parameters are enough to stat reliable statistical variables for VNSHmm and get improved performance.

### 4.2 Pinyin-to-Character Conversion

For the experiments of pinyin-to-character conversion, this paper adopts the same training corpus and test corpus as in pos tagging experiments. And 6763 Chinese frequent characters are chosen as the lexicon. This paper firstly converts all raw Chinese corpuses to the pinyin corpuses. Then based on the both kinds of corpuses, Hmm, NSHmm and VNSHmm are built up.

In the experiments, we first calculate KL distances between the state-transitionprobability distributions of Hmm and the distributions of NSHmm at different time. As we have done in the pos tagging experiments, we just calculate the average KL distance for each NSHmm. The results are presented in table 4.

**Table 4.** Average KL Distances between State-Transition-Probability Distributions of NSHmm

 and Hmm

Bin Number	K=1	K=2	K=3	K=4	K=5	K=6	K=7	K=8
Aver KL Dis	0	0.08	0.12	0.15	0.17	0.18	0.19	0.21

From table 4 we can see that as the bin number increases, the average KL distance become bigger and bigger and more predictive power is expected for NSHmm. And similar results can be gotten by comparing emission probability distributions of the two models. Then in the rest of this section, we perform the pinyin-to-character conversion experiments. Close test and open test are performed respectively. The results of close test are showed in table 5, figure 3 and the results of open test are presented in table 6, figure 4 respectively.

Table 5. Pinyin-to-Character Conversion Close Test

Bin Nu	mber	K=1	K=2	K=3	K=4	K=5	K=6	K=7	K=8
Hmm (ba	aseline)	8.30%							
NSHmm	Error Rate	8.30%	7.17%	6.55%	6.08%	5.74%	5.43%	5.19%	4.98%
	Reduction		13.61%	21.08%	26.75%	30.84%	34.58%	37.47%	40.00%
VNSHmm	Error	8.30%	8.28%	8.27%	8.28%	8.28%	8.28%	8.28%	8.28%
	Rate Reduction		0.24%	0.24%	0.24%	0.24%	0.24%	0.24%	0.24%



Fig. 3. Pinyin-to-Character Conversion Close Test

Table 6. Pinyin-to-Character Conversion Open Test

Bin Number		K=1	K=2	K=3	K=4	K=5	K=6	K=7	K=8
Hmm (ba	seline)	14.97%							
NSHmm	Error Rate	14.97%	12.62%	13.16%	13.61%	13.93%	14.23%	14.52%	14.81%
	Reduction		15.70%	12.09%	9.08%	6.95%	4.94%	3.01%	1.07%
VNSHmm	Error Poto	14.97%	11.98%	11.96%	11.96%	11.96%	11.97%	11.97%	11.97%
	Reduction		19.97%	20.11%	20.11%	20.11%	20.04%	20.04%	20.04%



Fig. 4. Pinyin-to-Character Conversion Open Test

In the experiments of pinyin-to-character conversion, the results are very similar to those in the pos tagging experiments. NSHmm and VNSHmm show much more predictive power than Hmm does. NSHmm gets at most 40% error rate reduction and VNSHmm obtains 0.24% reduction in close test; and they achieve 15.7% and 20.11% reductions respectively in open test. As time information increases, the error rate of NSHmm decreases drastically in close test as it dose in pos tagging task. And the overfitting problem arises after k=2 in open test.

However, different from the results of pos tagging experiments, VNSHmm outperforms NSHmm in open test. Since 6763 characters are adopted as states set in pinyin-to-character conversion system, which is much larger than the states set in pos tagging system, data sparseness problem is more likely to occur. VNSHmm can be view as a natural smoothing technique for NSHmm. Thus it works better. We also notice that the improvements in pinyin-to-character conversion experiments are more significant than those in pos-tagging experiments. In pinyin-to-character conversion task, the state chain is the Chinese sentence. Intuitively, some Chinese characters and words are much more likely to occur at some certain positions in the sentence, for instance, the beginning or the end of a sentence. As we discuss in section 3.3, in practice the time information of events in NSHmm is defined as the position information where the events occur. Then NSHmm and VNSHmm can capture those characteristics straightforwardly. But in pos-tagging, the state chain is the pos tag stream. Pos is a more abstract concept than word, and their positional characteristics are not as apparent as words'. Henceforth, the improvements in pos-tagging experiments are less significant than those in pinyin-to-character conversion experiments. But NSHmm and VNSHmm can still model and make good use of those positional characteristics, and notable improvements have been achieved.

In a word, NSHmm and VNSHmm achieve much lower error rates in both of the two sequence labeling tasks and show much more predictive power than Hmm.

# 5 Conclusions

To improve Hmm's predictive power and meanwhile avoid the problems of highorder Hmm, this paper relaxes the stationary hypothesis of Hmm, makes use of time information and proposes NSHmm. Moreover, to further reduce NSHmm's parameter space for mobile applications, VNSHmm is proposed by constructing statistical variables on the time information of NSHmm. Then NSHmm and VNSHmm are applied to two sequence labeling tasks: pos tagging and pinyin-to-character conversion. From the experiment results, we can draw three conclusions in this paper:

- Firstly, NSHmm and VNSHmm achieve much lower error rates than Hmm in both of the two tasks and thus have more predictive power.
- Secondly, the improvement of NSHmm is due to the increasing time information.
- Lastly, a small number of parameters are enough to stat the statistical variables for VNSHmm.

# 6 Further Research

Since NSHmm is an enhanced Hmm, some problems of Hmm also exist in NSHmm. For example, data sparseness problem is arising as time information increases in NSHmm. Some smoothing algorithms should be designed to solve it in our further work. Also it's difficult to describe long distance constraint for NSHmm and further research should be devoted to this problem. To construct more compact NSHmm, proper prone techniques should be further studied and be compared with VNSHmm.

### Acknowledgements

This investigation was supported emphatically by the National Natural Science Foundation of China (No.60435020) and the High Technology Research and Development Programme of China (2002AA117010-09).

We especially thank the three anonymous reviewers for their valuable suggestions and comments.

# References

- 1. F. Jelinek. Self-Organized Language Modeling for Speech Recognition. IEEE ICASSP, 1989.
- 2. George Nagy. At the Frontier of OCR. Processing of IEEE. 1992, 80(7).
- 3. ZhiMing Xu, XiaoLong Wang, Kai Zhang, Yi Guan. A Post Processing Method for Online Handwritten Chinese Character recognition. Journal of Computer Research and Development. Vol.36, No. 5, May 1999.
- Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. The Mathematics of Statistical Machine Translation: Parameter Estimation. Computational Linguistics. 1992, 19(2).
- Liu Bingquan, Wang Xiaolong and Wang Yuying, Incorporating Linguistic Rules in Statistical Chinese Language Model for Pinyin-to-Character Conversion. High Technology Letters. Vol.7 No.2, June 2001, P:8-13
- 6. Christopher D. Manning and Hinrich Schutze. Foundation of Statistic Natural Language Processing. The MIT Press. 1999.
- Brown, Peter F., Vincent J. Della Pietra, Peter V. deSouza, Jenifer C. Lai, and Robert L. Mercer. Class-based n-gram models of natural language. Computational Linguistics, 18(4):467-479. 1992.
- Z. Ghahramani and M. Jordan. Factorial hidden Markov models. Machine Learning, 29, 1997.
- 9. J. Fritsch. ACID/HNN: A framework for hierarchical connectionist acoustic modeling. In Proc. IEEE ASRU, Santa Barbara, December 1997.
- 10. Goodman, J. A bit of progress in language modeling. Computer Speech and Language, 403-434. 2001.
- 11. http://www.icl.pku.edu.cn