Deep Markov Neural Network for Sequential Data Classification

Min Yang[1] **Wenting Tu**[1] **Wenpeng Yin**[2] **Ziyu Lu**[1]
[1]Department of Computer Science, The University of Hong Kong, Hong Kong
{myang,wttu,zylu}@cs.hku.hk
[2]Center for Information and Language Processing, University of Munich, Germany
wenpeng@cis.lmu.de

Abstract

We present a general framework for incorporating sequential data and arbitrary features into language modeling. The general framework consists of two parts: a hidden Markov component and a recursive neural network component. We demonstrate the effectiveness of our model by applying it to a specific application: predicting topics and sentiments in dialogues. Experiments on real data demonstrate that our method is substantially more accurate than previous methods.

1 Introduction

Processing sequential data is a significant research challenge for natural language processing. In the past decades, numerous studies have been conducted on modeling sequential data. Hidden Markov Models (HMMs) and its variants are representative statistical models of sequential data for the purposes of classification, segmentation, and clustering (Rabiner, 1989). For most aforementioned methods, only the dependencies between consecutive hidden states are modeled. In natural language processing, however, we find there are dependencies locally and at a distance. Conservatively using the most recent history to perform prediction yields overfitting to short-term trends and missing important long-term effects. Thus, it is crucial to explore in depth to capture long-term temporal dynamics in language use.

Numerous real world learning problems are best characterized by interactions between multiple causes or factors. Taking sentiment analysis for dialogues as an example, the topic of the document and the author's identity are both valuable for mining user's opinions in the conversation. Specifically, each participant in the dialogue usually has specific sentiment polarities towards different topics. However, most existing sequential data modeling methods are not capable of incorporating the information from both the topic and the author's identity. More generally, there is no sufficiently flexible sequential model that allows incorporating an arbitrary set of features.

In this paper, we present a Deep Markov Neural Network (DMNN) for incorporating sequential data and arbitrary features into language modeling. Our method learns from general sequential observations. It is also capable of taking the ordering of words into account, and collecting information from arbitrary features associated with the context. Comparing to traditional HMM-based method, it explores deeply into the structure of sentences, and is more flexible in taking external features into account. On the other hand, it doesn't suffer from the training difficulties of recurrent neural networks, such as the vanishing gradient problem.

The general framework consists of two parts: a hidden Markov component and a neural network component. In the training phase, the hidden Markov model is trained on the sequential observation, resulting in transition probabilities and hidden states at each time step. Then, the neural network is trained, taking words, features and hidden state at the previous time step as input, to predict the hidden states at the present time step. The procedure is reversed in the testing phase: the neural network predicts the hidden states using words and features, then the hidden Markov model predicts the observation using hidden states.

A key insight of our method is to use hidden states as an intermediate representation, as a bridge to connect sentences and observations. By using hidden states, we can deal with arbitrary observation, without worrying about the issue of discretization and normalization. Hidden states are robust with respect to the random noise in the observation. Unlike recurrent neural net-

Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Short Papers), pages 32–37, Beijing, China, July 26-31, 2015.

work which connects networks between consecutive time steps, the recursive neural network in our framework connects to the previous time step by using its hidden states. In the training phase, since hidden states are inferred by the hidden Markov model, the training of recursive neural networks at each time step can be performed separately, preventing the difficulty of learning an extremely deep neural network.

We demonstrate the effectiveness of our model by applying it to a specific application: predicting topics and sentiments in dialogues. In this example, the sequential observation includes topics and sentiments. The feature includes the identity of the author. Experiments on real data demonstrate that our method is substantially more accurate than previous methods.

2 Related work

Modeling sequential data is an active research field (Lewis and Gale, 1994; Jain et al., 2000; Rabiner, 1989; Baldi and Brunak, 2001; Kum et al., 2005). The paper proposed by Kum et al. (2005) describes most of the existing techniques for sequential data modeling. Hidden Markov Models (HMMs) is one of the most successful models for sequential data that is best known for speech recognition (Rabiner, 1989). Recently, HMMs have been applied to a variety of applications outside of speech recognition, such as handwriting recognition (Nag et al., 1986; Kundu and Bahl, 1988) and fault-detection (Smyth, 1994). The variants and extensions of HMMs also include language models (Guyon and Pereira, 1995) and econometrics (Garcia and Perron, 1996).

In order to properly capture more complex linguistic phenomena, a variety of neural networks have been proposed, such as neural probabilistic language model (Bengio et al., 2006), recurrent neural network (Mikolov et al., 2010) and recursive neural tensor network (Socher et al., 2013). As opposed to the work that only focuses on the context of the sequential data, some studies have been proposed to incorporate more general features associated with the context. Ghahramani and Jordan (1997) proposes a factorial HMMs method and it has been successfully utilized in natural language processing (Duh, 2005), computer vision (Wang and Ji, 2005) and speech processing (Gael et al., 2009). However, exact inference and parameter estimation in factorial HMMs is intractable, thus the learning algorithm is difficult to implement and is limited to the study of real-valued data sets.

3 The DMNN Model

In this section, we describe our general framework for incorporating sequential data and an arbitrary set of features into language modeling.

3.1 Generative model

Given a time sequence $t = 1, 2, 3, \ldots, n$, we associate each time slice with an observation (s_t, u_t) and a state label y_t. Here, s_t represents the sentence at time t, and u_t represents additional features. Additional features may include the author of the sentence, the bag-of-word features and other semantic features. The label y_t is the item that we want to predict. It might be the topic of the sentence, or the sentiment of the author.

Given tuples (s_t, u_t, y_t), it is natural to build a supervised classification model to predict y_t. Recurrent neural networks have been shown effective in modeling temporal NLP data. However, due to the depth of the time sequence, training a single RNN is difficult. When the time sequence length n is large, the RNN model suffers from many practical problems, including the vanishing gradient issue which makes the training process inefficient.

We propose a Deep Markov Neural Network (DMNN) model. The DMNN model introduces a hidden state variable H_t for each time slice. It serves as an intermediate layer connecting the label y_t and the observation (s_t, u_t). These hidden variables disentangle the correlation between neural networks for each sentence, but preserving time series dependence. The time series dependence is modeled by a Markov chain. In particular, we assume that there is a labeling matrix L such that

$$P(y_t = i | H_t = j) = L_{ij} \tag{1}$$

and a transition matrix T such that

$$P(H_{t+1} = i | H_t = j) = T_{ij} \tag{2}$$

These two equations establish the relation between the hidden state and the labels. On the other hand, we use a neural network model M to model the relation between the hidden states and the observations. The neural network model takes (H_{t-1}, s_t, u_t) as input, and predict H_t as its output. In particular, we use a logistic model to define the probability:

$$P(H_t = i|H_{t-1}, s_t, u_t) \propto \exp((w_h^i, \phi(H_{t-1})) + (w_u^i, \varphi(u_t)) + (w_s^i N(s_t) + b)) \quad (3)$$

The vectors w_h, w_u, w_s are linear combination coefficients to be estimated. The functions ϕ, φ and function N turn H_{t-1}, u_t and s_t into featurized vectors. Among these functions, we recommend choosing $\phi(H_{t-1})$ to be a binary vector whose H_{t-1}-th coordinate is one and all other coordinates are zeros. Both function φ and function N are modeled by deep neural networks.

Since the sentence s_t has varied lengths and distinct structures, choosing an appropriate neural network to extract the sentence-level feature is a challenge task. In this paper, we choose N to be the recursive autoencoder (Socher et al., 2011a), which explicitly takes structure of the sentence into account. The network for defining φ can be a standard fully connect neural network.

3.2 Estimating Model Parameters

There are two sets of parameters to be estimated: the parameters L, T for the Markov chain model, and the parameters $w_h, w_u, w_s, \varphi, N$ for the deep neural networks. The training is performed in two phases. In the first phase, the hidden states $\{H_t\}$ are estimated based on the labels $\{y_t\}$. The emission matrix L and the transition matrix T are estimated at the same time. This step can be done by using the Baum-Welch algorithm (Baum et al., 1970; Baum, 1972) for learning hidden Markov models.

When the hidden states $\{H_t\}$ are obtained, the second phase estimates the remaining parameters for the neural network model in a supervised prediction problem. First, we use available sentences to train the structure of the recursive neural network N. This step can be done without using other information besides $\{s_t\}$. After the structure of N is given, the remaining task is to train a supervised prediction model to predict the hidden state H_t for each time slice. In this final step, the parameters to be estimated are $w_h, w_u, w_s$and the weight coefficients in neural networks N and φ. By maximizing the log-likelihood of the prediction, all model parameters can be estimated by stochastic gradient descent.

3.3 Prediction

The prediction procedure is a reverse of the training procedure. For prediction, we only have the sentence s_t and the additional feature u_t. By equation (3), we use (s_1, u_1) to predict H_1, then use (H_1, s_2, u_2) to predict H_2. This procedure continues until we have reached H_n. Note that each H_t is a random variable. Equation (3) yields

$$P(H_t = i|s, u) = \sum_j P(H_t = i|s_t, u_t, H_{t-1} = j) \cdot P(H_{t-1} = j|s, u) \quad (4)$$

This recursive formula suggests inferring the probability distribution $P(H_t|s, u)$ one by one, starting from $t = 1$ and terminate at $t = n$. After $P(H_t|s, u)$ is available, we can infer the probability distribution of y_t as

$$P(y_t = i|s, u) = \sum_j P(y_t = i|H_t = j)P(H_t = j|s, u) = \sum_j L_{i,j} P(H_t = j|s, u) \quad (5)$$

which gives the prediction for the label of interest.

3.4 Application: Sentiment analysis in conversation

Sentiment analysis for dialogues is a typical sequential data modeling problem.The sentiments and topics expressed in a conversation affect the interaction between dialogue participants (Suin Kim, 2012). For example, given a user say that "I have had a high fever for 3 days", the user may write back positive-sentiment response like "I hope you feel better soon", or it could be negative-sentiment content when the response is "Sorry, but you cannot join us today" (Hasegawa et al., 2013). Incorporating the session's sequential information into sentiment analysis may improve the prediction accuracy. Meanwhile, each participate in the dialogue usually has specific sentiment polarities towards different topics.

In this paper, the sequential labels available to the framework include topics and sentiments. In the training dataset, topics are obtained by running an LDA model, while the sentiment labels are manually labeled. The feature includes the identity of the author. In the training phase, the hidden Markov model is trained on the sequential labels, resulting in transition probabilities and hidden states at each time step. Then, the recursive autoencoders (Socher et al., 2011a) is trained, taking words, the identity of the author and hidden state at the previous time step as input, to predict the hidden states at the present time step. The procedure is reversed in the testing phase: the neural network predicts the hidden states using words

and the identity of the author, then the hidden Markov model predicts the observation using hidden states.

4 Experiments

To evaluate our model, we conduct experiments for sentiment analysis in conversations.

4.1 Datasets

We conduct experiments on both English and Chinese datasets. The detailed properties of the datasets are described as follow.

Twitter conversation (Twitter): The original dataset is a collection of about 1.3 million conversations drawn from Twitter by Ritter et al. (2010). Each conversation contains between 2 and 243 posts. In our experiments, we filter the data by keeping only the conversations of five or more tweets. This results in 64,068 conversations containing 542,866 tweets.

Sina Weibo conversation (Sina): since there is no authoritative publicly available Chinese short-text conversation corpus, we write a web crawler to grab tweets from Sina Weibo, which is the most popular Twitter-like microblogging website in China[1]. Following the strategy used in (Ritter et al., 2010), we crawled Sina Weibo for a 3 months period from September 2013 to November 2013. Filtering the conversations that contain less than five posts, we get a Chinese conversation corpus with 5,921 conversations containing 37,282 tweets.

For both datasets, we set the ground truth of sentiment classification of tweets by using human annotation. Specifically, we randomly select 1000 conversations from each datasets, and then invite three researchers who work on natural language processing to label sentiment tag of each tweet (i.e., positive, negative or neutral) manually. From 3 responses for each tweet, we measure the agreement as the number of people who submitted the same response. We measure the performance of our framework using the tweets that satisfy at least 2 out of 3 agreement.

For both datasets, data preprocessing is performed. The words about time, numeral words, pronoun and punctuation are removed as they are unrelated to the sentiment analysis task.

[1]http://weibo.com

Dataset	SVM	NBSVM	RAE	Mesnil's	DMNN
Twitter	0.572	0.624	0.639	0.650	0.682
Sina	0.548	0.612	0.598	0.626	0.652

Table 1: Three-way classification accuracy

4.2 Baseline methods

To evaluate the effectiveness of our framework on the application of sentiment analysis, we compare our approach with several baseline methods, which we describe below:

SVM: Support Vector Machine is widely-used baseline method to build sentiment classifiers (Pang et al., 2002). In our experiment, 5000 words with greatest information gain are chosen as features, and we use the LibLinear[2] to implement SVM.

NBSVM: This is a state-of-the-art performer on many sentiment classification datasets (Wang and Manning, 2012). The model is run using the publicly available code[3].

RAE: Recursive Autoencoder (Socher et al., 2011b) has been proven effective in many sentiment analysis tasks by learning compositionality automatically. The RAE model is run using the publicly available code[4] and we follow the same setting as in (Socher et al., 2011b).

Mesnil's method: This method is proposed in (Mesnil et al., 2014), which achieves the strongest results on movie reviews recently. It is a ensemble of the generative technique and the discriminative technique. We run this algorithm with publicly available code [5].

4.3 Experiment results

In our HMMs component, the number of hidden states is 80. We randomly initialize the matrix of state transition probabilities and the initial state distribution between 0 and 1. The emission probabilities are determined by Gaussian distributions. In our recursive autoencoders component, we represent each words using 100-dimensional vectors. The hyperparameter used for weighing reconstruction and cross-entropy error is 0.1.

For each dataset, we use 800 conversations as the training data and the remaining are used for testing. We summarize the experiment results in

[2]http://www.csie.ntu.edu.tw/~cjlin/liblinear/

[3]http://nlp.stanford.edu/~sidaw

[4]https://github.com/sancha/jrae/zipball/stable

[5]https://github.com/mesnilgr/iclr15.

Table 1. According to Table 1, the proposed approach significantly and consistently outperforms other methods on both datasets. This verifies the effectiveness of the proposed approach. For example, the overall accuracy of our algorithm is 3.2% higher than Mesnil's method and 11.0% higher than SVM on Twitter conversations dataset. For the Sina Weibo dataset, we observe similar results. The advantage of our model comes from its capability of exploring sequential information and incorporating an arbitrary number of factors of the corpus.

5 Conclusion and Future Work

In this paper, we present a general framework for incorporating sequential data into language modeling. We demonstrate the effectiveness of our method by applying it to a specific application: predicting topics and sentiments in dialogues. Experiments on real data demonstrate that our method is substantially more accurate than previous methods.

References

Pierre Baldi and Søren Brunak. 2001. *Bioinformatics: the machine learning approach*. MIT press.

Leonard E Baum, Ted Petrie, George Soules, and Norman Weiss. 1970. A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. *The annals of mathematical statistics*, pages 164–171.

Leonard E Baum. 1972. An equality and associated maximization technique in statistical estimation for probabilistic functions of markov processes. *Inequalities*, 3:1–8.

Yoshua Bengio, Holger Schwenk, Jean-Sébastien Senécal, Fréderic Morin, and Jean-Luc Gauvain. 2006. Neural probabilistic language models. In *Innovations in Machine Learning*, pages 137–186. Springer.

Kevin Duh. 2005. Jointly labeling multiple sequences: A factorial hmm approach. In *Proceedings of the ACL Student Research Workshop*, pages 19–24. Association for Computational Linguistics.

Jurgen V Gael, Yee W Teh, and Zoubin Ghahramani. 2009. The infinite factorial hidden markov model. In *Advances in Neural Information Processing Systems*, pages 1697–1704.

René Garcia and Pierre Perron. 1996. An analysis of the real interest rate under regime shifts. *The Review of Economics and Statistics*, pages 111–125.

Zoubin Ghahramani and Michael I Jordan. 1997. Factorial hidden markov models. *Machine learning*, 29(2-3):245–273.

Isabelle Guyon and Fernando Pereira. 1995. Design of a linguistic postprocessor using variable memory length markov models. In *Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on*, volume 1, pages 454–457. IEEE.

Takayuki Hasegawa, Nobuhiro Kaji, Naoki Yoshinaga, and Masashi Toyoda. 2013. Predicting and eliciting addressee's emotion in online dialogue. In *ACL (1)*, pages 964–972.

Anil K Jain, Robert P. W. Duin, and Jianchang Mao. 2000. Statistical pattern recognition: A review. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(1):4–37.

Hye-Chung Monica Kum, Susan Paulsen, and Wei Wang. 2005. Comparative study of sequential pattern mining models. In *Foundations of Data Mining and knowledge Discovery*, pages 43–70. Springer.

Amlan Kundu and Paramrir Bahl. 1988. Recognition of handwritten script: a hidden markov model based approach. In *Acoustics, Speech, and Signal Processing, 1988. ICASSP-88., 1988 International Conference on*, pages 928–931. IEEE.

David D Lewis and William A Gale. 1994. A sequential algorithm for training text classifiers. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 3–12. Springer-Verlag New York, Inc.

Grégoire Mesnil, Marc'Aurelio Ranzato, Tomas Mikolov, and Yoshua Bengio. 2014. Ensemble of generative and discriminative techniques for sentiment analysis of movie reviews. *arXiv preprint arXiv:1412.5335*.

Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *INTERSPEECH*, pages 1045–1048.

R Nag, K Wong, and Frank Fallside. 1986. Script recognition using hidden markov models. In *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'86.*, volume 11, pages 2071–2074. IEEE.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 79–86. Association for Computational Linguistics.

Lawrence Rabiner. 1989. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286.

Alan Ritter, Colin Cherry, and Bill Dolan. 2010. Unsupervised modeling of twitter conversations.

Padhraic Smyth. 1994. Hidden markov models for fault detection in dynamic systems. *Pattern recognition*, 27(1):149–164.

Richard Socher, Jeffrey Pennington, Eric H Huang, Andrew Y Ng, and Christopher D Manning. 2011a. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 151–161. Association for Computational Linguistics.

Richard Socher, Jeffrey Pennington, Eric H. Huang, Andrew Y. Ng, and Christopher D. Manning. 2011b. Semi-Supervised Recursive Autoencoders for Predicting Sentiment Distributions. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Richard Socher, Alex Perelygin, Jean Y Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1631–1642. Citeseer.

Alice Oh Suin Kim, JinYeong Bak. 2012. Discovering emotion influence patterns in online social network conversations. In *SIGWEB ACM Special Interest Group on Hypertext, Hypermedia, and Web*. ACM.

Peng Wang and Qiang Ji. 2005. Multi-view face tracking with factorial and switching hmm. In *Application of Computer Vision, 2005. WACV/MOTIONS'05 Volume 1. Seventh IEEE Workshops on*, volume 1, pages 401–406. IEEE.

Sida Wang and Christopher D Manning. 2012. Baselines and bigrams: Simple, good sentiment and topic classification. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2*, pages 90–94. Association for Computational Linguistics.