Feature Learning for Conditional Random Fields and its Application to Gesture Recognition

Jie Liu *
Nankai University
jliu@soe.ucsc.edu

Kai Yu
NEC Laboratories America
kyu@sv.nec-labs.com

Yi Zhang
UC Santa Cruz
yiz@soe.ucsc.edu

Yalou Huang
Nankai University
yellow@nankai.edu.cn

Abstract

Conditional random fields (CRFs) have been successful in many sequence labeling tasks, which conventionally rely on a hand-craft feature representation of input data. However, a powerful data representation could be another determining factor of the performance, which has not attracted enough attention yet. We describe a novel sequence labeling framework for gesture recognition, which builds a supervised CRF and an unsupervised dynamic model on a shared nonlinear feature transformation neural network. The model is a case of transfer learning that jointly optimizes two learning tasks together with learning a meaningful feature representation of input data. We demonstrate a gesture recognition system that yields a significant improvement of recognition accuracy over conventional CRFs.

1 Introduction

Motivated by the desire for improved human-machine communication systems, recognizing visual gestures has been studied by many researchers recently. Due to its dynamic an dependence between observations over time, gesture recognition has been formulated into a sequence labeling problem which arises from many scientific fields. There are many works on recognizing head nod [1], eye gaze behavior [2] [3] [4] and arm gestures [5]. As a well understood and widely used model, Hidden Markov models(HMM) [6] have been popular in gesture recognition and general human action recognition systems. However, HMMs are limited by its strict assumption on the conditional independence of input data.

It is widely recognized that the Conditional Random Fields [7] models are very powerful in sequence learning. CRF have been applied successfully to many sequence labeling tasks such as Natural Language Processing [8] [9] [10] [11], bioinformatics [12] [13], and computer vision [14] [15] etc., because they have good ability of learning the structures and dynamics of sequence data.

However, it is also important to capture the intrinsic representation of the observations of sequences, on which there is not much attention has been paid. Impropriate features result in bad performance, even though good models are used. Designing features manually is hard for people and could consume much time on tuning, which could be regarded as a important part of the cost of a learning task, especially when the problem is complex. How to learn features from data automatically is getting increasingly interest. The success of other techniques, such as Singular Value Decomposition and Principle component analysis, demonstrates the powerful of learning hidden representation of the

*Work done while the author was visiting UC Santa Cruz.

observations. Unfortunately, little research attention has been paid on learning the hidden representation in the context of sequence modeling. Besides, data labeling is a widely recognized cost, and how to learn a sequence model with limited training data is a challenging problem. Many previous works have been done to solve such problem in tradition i.i.d machine learning tasks. Models that can utilize unlabeled data have been proposed in recent years.

Motivated by prior work in non sequence modeling tasks, we propose a transfer learning framework for the sequence models, especially CRF. Our approach is to learn powerful hidden features automatically and utilizing unlabeled sequence data, while retaining the ability of capturing dynamics of CRF. More specifically, a transformation layer is added to a CRF model to capture the hidden representation of the input raw vector. As a component of the transfer learning framework, we design an auxiliary learning task to share the transformation parameters, which give the model much better ability of generalization. Furthermore, the auxiliary learning task is a unsupervised generative language model with a similar transformation layer. This enables us to make use of large amount of unlabeled sequence data. We evaluate the algorithm on continuous gesture data. The performance of our model is much better than standard CRF under both supervised and semi-supervised learning condition.

2 Transfer Learning for Conditional Random Fields

The topic of transfer learning is also referred as *multi-task* learning or *learning to learn*, motivated by the fact that many different learning tasks are related in some way and a communication of knowledge between them may improve the performances of all the tasks. For example, for human, knowledge of one language is helpful for learning another Language.

In this section, we first introduce some basic of Conditional Random Field model, and then describe how our transfer learning sequence model are built. There are two learning tasks in the model. One is the main task, which is a CRF mounted on a hidden representation layer. The other is an unsupervised sequence model sharing the layer. Communicating through the shared parametric layer, these two learning tasks are learned from the same dataset $\mathcal{D} = \{\mathbf{x}^{(n)}, \mathbf{y}^{(n)}\}_{n=1}^{N}$ simultaneously, where $\mathbf{x}^{(n)}$ is a sequence of observation labeled as sequence $\mathbf{y}^{(n)}$.The relationship between the tasks are the set of parameters shared by all the learning tasks.

2.1 Condition Random Fields

CRFs have been very successful in many sequence labeling tasks. By modeling conditional probability $P(\mathbf{y}|\mathbf{x})$ rather than joint probability $P(\mathbf{y}|\mathbf{x})$, the discriminative models are believed better than generative models [16]. In addition, the CRF models allow arbitrary, non-independent features on the observation sequence X, because the probability of a transition between labels may depend on past and future observations. For a CRF model, we want to learn a mapping from a observation sequence $\mathbf{X} = \{x_1, x_2, ..., x_T\}$ to a label sequence $\mathbf{Y} = \{y_1, y_2, ..., y_T\}$. A CRF defines the conditional probability of $y^{(n)}$ as

$$\begin{aligned} P(\mathbf{y}|\mathbf{x};\theta) &= \frac{1}{Z(\mathbf{x})}\prod_t \Phi(y_t, x_t;\theta) \\ &= \exp\left(\theta^\top \mathbf{F}(\mathbf{y},\mathbf{x}) - \log Z(\mathbf{x};\theta)\right) \end{aligned} \tag{1}$$

where $\mathbf{F}(\mathbf{y},\mathbf{x}) = \sum_t f(\mathbf{y}_t, \mathbf{x}, t)$, $Z(\mathbf{x})$ is a normalization factor over all the possible states of $\mathbf{y}'$ of length $|\mathbf{x}|$ defined as

$$Z(\mathbf{x}) = \sum_{\mathbf{y}'} \exp\left(\sum_t \Phi(y'_t, x_t;\theta)\right) \tag{2}$$

and Φ is a parametric potential function

$$\Phi(y_t, x_t;\theta) = \theta^\top f(y_t, \mathbf{x}, t) \tag{3}$$

where θ is a vector of linear weights and $f(y_t, \mathbf{x}, t)$ computes a set of features given the node at time t.

Typically, given a set of training example $\mathcal{D} = \{\mathbf{x}^{(n)}, \mathbf{y}^{(n)}\}_{n=1}^{N}$, where $x^{(n)} \in \mathcal{X}$ and $y^{(n)} \in \mathcal{Y}$, for example, video sequences of gesture, the linear weights can be estimated by maximizing the penalized log-likelihood with respect to the conditional probability

$$\max_{\theta} \left\{ \sum_{n=1}^{N} \log P(\mathbf{y}^{(n)} | \mathbf{x}^{(n)}; \theta) \right\} \tag{4}$$

There are efficient exact inference algorithms for linear chains CRFs.

2.2 CRF with Hidden Features

In this subsection, we describe a CRF model that is capable of learning powerful feature from data automatically. To make the CRF have the ability to learn features as well as transfer learning, we introduce a nonlinear transformation layer to compute hidden representation of input observation vectors. Here the hidden representation can be viewed as powerful features leaned from input data. Consider a sequence of observations $\mathbf{x} = \{x^{(n)}\}_{i=1}^{N}$ and labels $\mathbf{y} = \{y^{(n)}\}_{i=1}^{N}$. Let $y_t \in y = \{1, \cdots, C\}$, we use $c_t = [\mathbb{I}_{n=y_t}]_{n=1}^{N}$ to encode y_t, e.g., if $y_t = 2$ and $C = 4$, then $c_t = [0, 1, 0, 0]^T$, we define our linear-chain CRF as

$$\begin{aligned} P(\mathbf{y}|\mathbf{x}; \theta, \alpha) &= \frac{1}{Z(\mathbf{x}; \theta)} \exp \left\{ \sum_{t} \theta^{\top} f(y_t, \phi(\mathbf{x}_t; \alpha), t) \right\} \\ &= \frac{1}{Z(\mathbf{x}; \theta)} \exp \left\{ \sum_{t} \langle \lambda, c_t \otimes c_{t-1} \rangle + \langle \mu, c_t \otimes \phi(\mathbf{x}_t; \alpha) \rangle \right\} \end{aligned} \tag{5}$$

where $\otimes$ is Kronecker product, $\theta = \{\lambda, \mu\}$, and $\mathbf{x}_t = \{x_i | i = t - \frac{Q-1}{2}, \cdots, t + \frac{Q-1}{2}\}$ is segment of length Q centered at t of $\mathbf{x}$. In its simplest form, $\phi(\mathbf{x}_t; \alpha)$ is a $Q \times M$ vector by concatenating $\{\phi(x_i; \alpha) | x_i \in (x)_t\}$, where $\phi(x_i; \alpha)$ is a nonlinear function $x \to \mathbb{R}^M$ with parameter α. Then λ is a $C \times C$ parameter vector, and μ is a $C \times Q \times M$ parameter vector. There are many other ways to enrich the model, for example, by defining highter-order features $c_{t-1} \otimes c_t \otimes \phi(\mathbf{x}_t; \alpha)$.

The proposed model (5) is like a multi-layer neural network (NN), which generally optimizes the classifier and hidden-layer features simultaneously. Our model, as is shown in Figure **??**, puts a CRF upon $\phi(x_i; \alpha)$ that amounts to a hidden layer. In fact, our implementation $\phi(x_i; \alpha)$ itself is an neural network

$$\phi_i(x_i; \alpha) = \varrho \left(\sum_{k=1}^{H} \omega_{i,k}^{\phi} h_k(x) + b_i^{\phi} \right), \quad h_k(x) = \varrho \left(\sum_{j=1}^{D} \omega_{k,j}^{h} x_j + b_k^{h} \right) \tag{6}$$

where $i = 1, \cdots, M$, ϱ is an non-linear (tanh) transfer function, and the parameters α include all the weights ω and bias term b. Therefore, the overall CRF-NN bybrid contains two nonlinear hidden layers and one structured-output layer.

The model (5) is different with traditional CRF (1) in its learning process. It not only learns the linear parameters, e.g., λ and ω, but also optimizes the features by tuning the transformation parameter α in transformation function ϕ. This capacity of learning features is important, especially when the task if complex.

2.3 Generative Sequence Model with Hidden Feature

In CRF model, the transition between different labels are captured to learn the model. Besides such transitions, in structure data, there is some relationship between the neighbor nodes of the observation sequence, which can be caputured regardless the labels. Taking the sequence of human action as a example, the action at the time t usually is an extension of the trend of action in the past several time points, that is there is some physical dependence between consecutive observations in a sequence. As a second task, we design a sequence model to learn such dependence between neighbor observation nodes over a sequence. Such a model is like a statistical language model. It can be represented by the conditional probability of the next observation given all the previous ones, since

$$P(\mathbf{x}) = \prod_{t=1}^{T} P(x_t | x_{1,\cdots,t-1}), \tag{7}$$

where x_t is the t-th observation vector, and writing sub-sequence $x_{i,\cdots,j} = (x_i, x_{i+1}, \cdots, x_{j-1}, x_j)$. Such statistical language models have already been found useful in many technological application.

Taking the advantage of observation vector order can considerably reduce the difficulty of this modeling problem. The fact that temporally closer vectors in the sequence are statistically more dependent. Thus, $n\text{-}gram$ models construct tables of conditional probabilities for the next vector:

$$P(x_t | x_1^{t-1}) \approx P(x_t | x_{t-n+1,\cdots,t-1}). \tag{8}$$

With the learned feature output by the NN, the 1-$gram$ language model is

$$P(\mathbf{x}; \theta) = \prod_{t} P(x_t | x_{t-1}; \theta) \tag{9}$$

We use a $softmax$ function as the output layer, if we regard the whole model as neural sequence model.

$$P(x_t | x_{t-1}; \alpha, \mathbf{A}) = \frac{\exp \Psi(\phi_t, \phi_{t-1}; \mathbf{A})}{\int \exp \Psi(\phi_t, \phi'_{t-1}; \mathbf{A}) d\phi'_{t-1}} \tag{10}$$

where Ψ is the potential function, $\mathbf{A}$ is a parameter matrix, and the function ϕ is exactly the nonlinear function used by neural CRF.

It can be very flexible to design the potential functions for the learning tasks under different application scenarios. In our experiment for gesture recognition, we use two functions. First, (11) is used to learn the dependence between the nodes of the sequence.

$$\Psi(\phi_t, \phi_{t-1}; \boldsymbol{A}) = -||\boldsymbol{A} \cdot \phi_{t-1} - \phi_t||^2. \tag{11}$$

From the (5) and (9), we can see that both the tasks share the same transformation ϕ. By sharing the same transformation, we can achieve a better generalization. In other words, the knowledge learned from other task can improve the new learning task by transferring its shared part of knowledge. Then we can use shared parameter of the transformation function to capture the common knowledge of the tasks. The input vector can be transformed to flexible dimensions. The result vector of transformation, ϕ_t, can be regarded as new feature learned by the model.

Besides the benefit on the generalization, the sequence model can be learned from unlabeled data as well as from labeled data. Together with the supervised learning model in the first task, the whole training process is semi-supervised learning.

3 Joint Training of Labeled & Unlabeled Sequence Models

With the supervised model (5) and the unsupervised model (9), it is natural to consider the joint model $P(\mathbf{y}, \mathbf{x}; \theta) = P(\mathbf{y}|\mathbf{x}; \theta) \prod_t P(\mathbf{x}_t | \mathbf{x}_{t-1}; \theta)$. Note that, without sharing of parameters, the two models are independent. The loss function for the two tasks are defined as (12) and (13) respectively.

$$\ell_1(\lambda, \mu, \alpha) = \sum_{n=1}^{N} -\log P(\mathbf{y}^{(n)} | \mathbf{x}^{(n)}; \lambda, \mu, \alpha). \tag{12}$$

$$\ell_2(A, \alpha) = \sum_{n=1}^{N} \frac{1}{T} \sum_{t} -\log P(x_t^{(n)} | x_{t-1}^{(n)}; \alpha, A) \tag{13}$$

Letting the two models share the nonlinear feature transformation $\phi(x; \alpha)$, we are able to develop a transfer learning framework for sequence model: Given labeled sequences $\{\mathbf{x}^{(n)}, \mathbf{y}^{(n)}\}$, $n = 1, \cdots, N$, the joint model aims to minimize the loss

$$\ell(\theta) = \ell_1(\lambda, \mu, \alpha) + \eta \ell_2(\alpha, A) + \frac{1}{2\sigma^2} ||\theta||^2 \tag{14}$$

where parameter vector $\theta = (\lambda, \mu, A, \alpha)$. The third term is the log of a Gaussian prior with variance σ^2, i.e., $P(\theta) \sim \exp(\frac{1}{2\sigma^2}||\theta||^2)$.

We use the gradient descendent method to search for the optimal parameters. For the second task, the derivatives are pretty straightforward to be calculated, so we omit them for lack of space. Below are some derivatives from the supervised neural CRF:

$$\frac{\partial \ell_1}{\partial \lambda_i} = \sum_i f_i(y_t, \phi_t, t) - \sum_t \sum_{y'_t} f_i(y'_t, \phi_t, t) P(y'_t|x) \tag{15}$$

$$\frac{\partial \ell_1}{\partial \alpha_{i,j}} = \sum_t \lambda_i \frac{\partial f_i(y_t, \phi_t, t)}{\partial \alpha_{i,j}} - \sum_t \sum_{y'_t} \lambda_i \frac{\partial f_i(y'_t, \phi_t, t)}{\partial \alpha_{i,j}} P(y'_t|x) \tag{16}$$

Note that the marginal probabilities $P(y_t|\mathbf{x})$ can be computed efficiently using belief propagation. In our experiments, we use BFGS to optimize the objective function.

3.1 Training with Unlabeled Data

Labeling data can be very expensive, while the unlabeled data are usually abundant and easy to get. With the unsupervised sequence model, it is very straightforward to use unlabeled data when training. Given labeled sequences $\{\mathbf{x}^{(n)}, \mathbf{y}^{(n)}\}, n = 1, \cdots, L$, and unlabeled sequence $\{x^{(n)}\}, n = L+1, \cdots, L+U$, we get a semi-supervised sequence model minimizing the similar loss with (14), while the ℓ_1 and ℓ_2 are slightly changed:

$$\ell_1(\lambda, \mu, \alpha) = -\frac{1}{L} \sum_{n=1}^{L} \log P(\mathbf{y}^{(n)}|\mathbf{x}^{(n)}; \lambda, \mu, \alpha) \tag{17}$$

$$\ell_2(\boldsymbol{A}, \alpha) = -\frac{1}{L+U} \sum_{n=1}^{L+U} \log P(\mathbf{x}^{(n)}; \boldsymbol{A}, \alpha) \tag{18}$$

In the joint training of the two learning task, the same gradient descendent optimization technique, BFGS, is performed to search the optimal parameters.

4 Inference

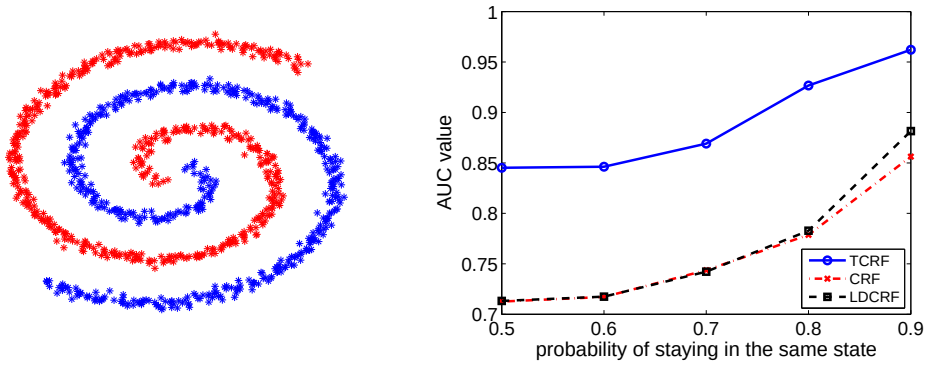
The main goal of inference is to infer the most probable label, given the values of the observed nodes. Given the model parameters θ^* learned from training data, prediction of a new test sequence X is to estimate the label sequence Y^* with the maximum probability

$$Y^* = \arg\max_Y P(Y|X, \theta^*) \tag{19}$$

There are two main methods to estimate the labels. One is compute the Viterbi path, the other is to compute the maximum marginal probabilities. In our experiments, we use the latter to minimize the err per frame. As discussed in previous subsection, the marginal probability can be computed efficiently using belief propagation.

5 Experimental Results

In this section, we evaluate the performance of the model on visual gesture recognition task, a specific example of sequence labeling problem. We choose this task because it is an important problem that arises from many scientific fields, and because discriminative sequence models, like CRF and Latent Dynamic Conditional Field (LDCRF) [17], have achieved good performance on the tasks.



(a) (b)

Figure 1: Result on synthetic data. (a): The data is comprised of two class of spiral data. (b): The evolution of the auc values over the increase of the transition probabilities inside each class.

5.1 Synthetic Experiment

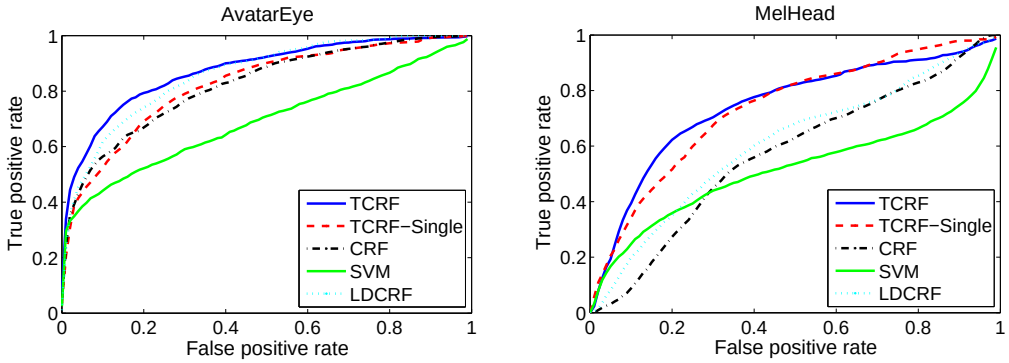
We will first perform experiments on a synthetic data to show the improvement offered by capability of our framework in learning features. The synthetic data consists of two spiral data shown in Figure (1). A set of sequences are sampled according to an HMM with two state, where each states emits instances uniformly from one of the class. Let p is the probability of staying in the same class and $1-p$ is the chance of transiting to the other class. The idea is to show the improvement offer by the learned hidden feature that the smaller the p is, the dependence between neighboring labels. In other words, the feature function measuring the dependence of labels contributes less with the decrease of p.

We sample 200 sequences of length 100 with different p. Sequence model with feature learning (TCRF) and sequence model without feature learning (CRF and LDCRF). The values on x-axis are the probabilities of staying in the same state and y-axis is the AUC (the area under the ROC curve) value. Figure 1 (b) shows that the TCRF outperforms the other two sequence models a steady margin with p varies from 0.5 to 0.9. When p is 0.5, the states transit to each other totally randomly, which directs the models into i.i.d. classification tasks. Under such condition, we can say that the improvement of the TCRF is offered by the feature learning.

5.2 Gesture Recognition

Two visual gesture recognition datasets are used in our experiments. One is **AvatarEye**: This dataset consists of eye gaze estimates from 6 human participants interaction with a virtual embodied agent. The goal is to recognize gaze aversion gestures from unsegmented video sequences. Eache video sequence lasted approximately 10-12 minutes, and was recorded at 30 frames/sec. Each frame was labeled as a gaze-aversion gesture or a background gesture/motion. The other is **MelHead**: This dataset consists of head velocities from 16 human participants. The gestures were video recorded and labeled start and end point of each head nod. There are total 274 head nods were performed by the participants.

Both the datasets are very unbalanced because of the large amount background frames. In order to train the models efficiently, we randomly remove some of the background frames in the training sequences. The resulted training sequences consisted of equal numbers of sequences with and without target frames. The sequence with target frames are the subsequences of the consecutive positive labeled frames and the buffers before and after it. The length of buffers varied from 2 to 50 randomly. The pure background sequences are randomly cut from the sequences with length vary from 30 to 60. To compare and demonstrate the performance, we employ receiver operation characteristic (ROC) curve and the area under the ROC curve (AUC) [18].



(a) (b)

Figure 2: Result under supervised learned model

On these two gesture recognition tasks, we describe our evaluation of the performance of our proposed model against the baseline models. In addition to supervised learning experiments, we present the performance of our model under semi-supervised leaning process as well.

In the supervised learning experiments, we performed K-fold testing approach where k sequences were hold out for testing while all other sequences for training and validation. N/k times of training and testing are performed, where N is the number of total sequence of the dataset. K are 4 and 1 for the Melhead and Avatareye respectively.

Our proposed transfer learning CRF (TCRF) was compared with standard CRF, Latent-Dynamic CRF (LDCRF), and Support Vector Machine (SVM). Besides, we also trained our model without the second task model(describe in sec. 2.3), which is noted as TCRF-Single.

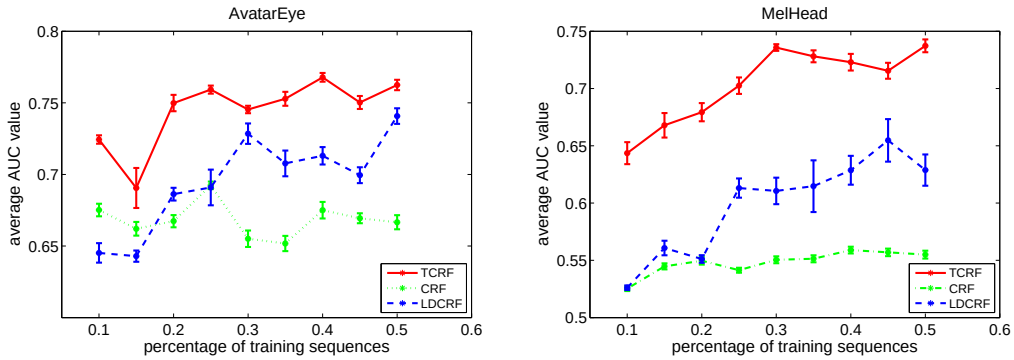
In the experiments we compared our transfer learning Conditional Random Field (TCRF) model with standard Conditional Random Field (CRF), Latent Dynamic Conditional Field (LDCRF), and Support Vector Machine (SVM) with a linear kernal. The output of the models when testing are the probabilities of the possible labels of each frame.

As is shown in Figure 2, our framework of transfer learning for CRF yielded the best performance [1]. It is noteworthy that the performance of TCRF on single task is also better than the standard CRF, which indicated that only learning hidden representation itself without the support of the second task still can be helpful.

Under the semi-supervised learning setting, we divided the dataset into k folds where k sequences were hold out for testing as in the supervised learning experiments. On each fold, we trained the models using ascendent percentage of the training sequences as labeled data while the rest of the training sequences were used as unlabeled data. The random selection of the labeled data were performed m times of each percentage, that is, the performance value at each amount of labeled data is an average of m$\times$k results. For AvatarEye and Melhead, k was 3 and 4 respectively, and n was 5 for both dataset.

As is shown in the Figure 3, our proposed frame work of sequence model outperform the supervised models like CRF and LDCRF a large margin on both dataset. On AvatarEye dataset, the LDCRF showed better performance when labeled sequences are more than 20%, and catched up TCRF with the increase the amount of labeled sequences. For the Melhead dataset, there is less improvement offered by the increase of the amount of labeled sequence. This phenomenon may caused by the different size of the two dataset. The AvatarEye dataset had only 6 participants and 77 eye gestures, while the MelHead is much bigger (274 head nods). So under same percentage, the sequences of MelHead provided more knowledge to the models to learn the manifold, which left less room for improvement when more labeled sequences were added. However, our proposed model demonstrated a significant superior performance when little labeled sequences were used.

[1] Although LDCRF model didn't reproduce as good performance as in [17] due to different preprocessing of training data, it is consistent that it achieved better performance on both dataset than CRF.



(a) (b)

Figure 3: Result under semi-supervised learned model. The values on x axis are the the percentages of the training sequences used as labeled data, the values on y axis are the evaluation metric, which is AUC here.

6 Conclusions

There are limited research on transfer learning for structure learning. In this paper, we proposed a framework for performing transfer learning on sequence model. With shared parameters, our model allows the multiple tasks to robustly communicate information from each other, but powerful hidden features are learned automatically. Additionally, enjoying the ability of the generative sequence model to make use unlabeled data, the proposed framework can smoothly turn to a semi-supervised learning method. Experimental results have demonstrated that our proposed model outperforms the standard CRF a large margin and is comparable with the state-of-the-art model LDCRF on this recognition task under supervised learning setup, which suggested that the learned hidden features and communication between learning tasks offer significant benefits. The semi-supervised learning experiments have also showed superior performance of our proposed model, which is an another important contribution against other existing models.

References

[1] A. Kapoor and R. Picard. A real-time head nod and shake detector, 2001.

[2] A. Colburn, M. Cohen, and S. Drucker. The role of eye gaze in avatar mediated conversational interfaces, 2000.

[3] Atsushi Fukayama, Takehiko Ohno, Naoki Mukawa, Minako Sawaki, and Norihiro Hagita. Messages embedded in gaze of interface agents — impression management with agent's gaze. In *CHI '02: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 41–48, New York, NY, USA, 2002. ACM Press.

[4] Gaspard Breton, Danielle Pelé, Christophe Garcia, Franck Panaget, and Philippe Bretier. Modeling gaze behavior for a 3d eca in a dialogue situation. pages 252–255. 2006.

[5] M. Brand, N. Oliver, and A. Pentland. Coupled hidden markov models for complex action recognition. pages 994–999, 1997.

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

[7] John Lafferty, Andrew Mccallum, and Fernando Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. 18th International Conf. on Machine Learning*, pages 282–289. Morgan Kaufmann, San Francisco, CA, 2001.

[8] Ben Tasker, Abbeel Pieter, and Daphne Koller. Discriminative probabilistic models for relational data. In *Proceedings of the 18th Annual Conference on Uncertainty in Artificial Intelligence (UAI-02)*, pages 485–49, San Francisco, CA, 2002. Morgan Kaufmann.

[9] Fuchun Peng and Andrew Mccallum. Information extraction from research papers using conditional random fields. *Information Processing & Management*, 42(4):963–979, July 2006.

[10] Burr Settles. Abner: an open source tool for automatically tagging genes, proteins, and other entity names in text. *Bioinformatics*, April 2005.

[11] F. Sha and F. Pereira. Shallow parsing with conditional random fields, 2003.

[12] Kengo Sato and Yasubumi Sakakibara. RNA secondary structural alignment with conditional random fields. *Bioinformatics*, 21(Suppl. 2):ii237–ii242, 2005.

[13] Yan Liu, Jaime Carbonell, Peter Weigele, and Vanathi Gopalakrishnan. Segmentation conditional random fields (scrfs): A new approach for protein fold recognition.

[14] Xuming He, R. S. Zemel, and M. A. Carreira-Perpinan. Multiscale conditional random fields for image labeling. volume 2, pages II–695–II–702 Vol.2, 2004.

[15] S. Kumar and M. Hebert. Discriminative fields for modeling spatial dependencies in natural images, 2003.

[16] Vladimir N. Vapnik. *The Nature of Statistical Learning Theory (Information Science and Statistics)*. Springer, November 1999.

[17] Louis-Philippe Morency, Ariadna Quattoni, and Trevor Darrell. Latent-dynamic discriminative models for continuous gesture recognition. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pages 1–8, 2007.

[18] J. A. Hanley and B. J. McNeil. The meaning and use of the area under a receiver operating characteristic (roc) curve. *Radiology*, 143(1):29–36, April 1982.