2012 Eighth International Conference on Computational Intelligence and Security

Analyzing Sequence Data Based on Conditional Random Fields with Co-training

Leilei Yang†,Guiquan Liu‡*,Qi Liu†,Lei Zhang†,Enhong Chen‡
University of Science and Technology of China
†{*yangll,feiniaol,stone*}*@mail.ustc.edu.cn*,‡{*gqliu,cheneh*}*@ustc.edu.cn*

***Abstract*—Sequence data plays an important role in data analysis applications, such as sequence classification. One important aspect of sequence data analysis is to obtain the labeled sequence data and use a machine learning model to predict the sequence structures. Conditional Random Fields (CRF) is such a machine learning method which is popular used in sequential data analysis. This is because that CRF can effectively capture the data correlations in context with abundant training data. However, in real applications, the labeled training data is usually difficult to be collected. In order to reduce the requirement of the amount of the labeled training data, a novel model is proposed named Conditional Random Fields with Co-training (Co-CRF). The Co-CRF model can work well even on the reduced labeled training data. Empirical results show that Co-CRF can produce a more accurate analysis than the traditional CRF, especially with very limited training data.**

***Keywords*-Sequence Data; Conditional Random Fields; Co-training; Classification;**

I. Introduction

In applications, the correlations and patterns among the statistical data are often difficult to be mined in cross-section scenarios. Comparatively, the correlations extracted from sequences are more informative. Thus, sequence data are much useful in data presentations, and actually they (especially the time-series data) have been widely observed and collected. Therefore, sequence data plays an important role in data analysis applications, such as sequence classification.

In order to analyze the sequence data, some traditional probabilistic models are proposed, such as *Hidden Markov models (HMMs)* [9] and *Maximum Entropy Model* [4]. Whereas these models always treat atomic elements in sequence as processing units. This strategy may lose some information about the correlations among data in context. These models entrapped themselves in the Label Biased Problem [6], which drives them to be biased towards states with few successor states. And even worse, these generative models must make very strict independence assumptions on the observations.

The drawbacks of the strict independence assumptions motivate to looking for conditional models. To this end, *Conditional Random Fields (CRF)* [5, 6, 10, 12] is proposed to construct a discriminative conditional models. It is often applied in pattern recognition and machine learning, where it is treated as a structured prediction [11] tool. Under a discriminative framework, CRF constructs a conditional model from paired observation and label sequences, rather than a joint probability in generative models. This structure reduces the modeling time, since it avoids enumerating all possible observation sequences. As the ordinary classifiers predict a label for a single sample ignoring the neighbor samples, taking context information into account is beneficial to improving the accuracy of the CRF. Therefore, it is often used for labeling or parsing of sequence data, such as natural language text or biological sequences, etc. In this paper, the CRF serves as a classifier to label the elements in sequence. The CRF offers several advantages in building probabilistic models to analyze sequences, including the ability to relax strong independence assumptions which are often assumed in traditional methods. Furthermore, the CRF treats the data collected in a period of time as a processing unit. This prevents the CRF from losing information about correlations among data.

In real applications, abundant labeled data is required to build a sequence analytical model (e.g. CRF) precisely. Whereas, the training data is generally difficult to be collected. Thus *Co-training* [1, 7] is introduced into the framework for training a good CRF classifier with limited amount of training data. Co-training is a kind of semi-supervised learning method. It is applied in this work to reduce the impact of limited training data. Co-training assumed that features can be split into two sets. And each sub-feature set is sufficient to train a good classifier independently. Therefore the two sets are demanded to be independent. However, it is impossible to obtain the completely independent feature sets. We relax the independent assumption in applications. So the splitting method is focus on weakening the linear independence between the sub-feature sets. Finally, the empirical results show that our solution is good enough to address the linear correlation in sequence processing.

Based on the theoretical support presented above, *Conditional Random Fields with Co-training (Co-CRF)* is proposed to the framework which combines the CRF with Co-training structure. In this way we overcomes many traditional problems while classify a processing unit. Specifically, the Co-training structure first splits up the attributes of the data so as to construct two separate CRF classifiers. Then the two classifiers select the most confident samples for the other one's training process. This interactive procedure reduces the required amount of labeled training data. Thus, given the

*Contact Authors.

978-0-7695-4896-8/12 $26.00 © 2012 IEEE
DOI 10.1109/CIS.2012.29

same training data, the Co-CRF could improve the accuracy of the model effectively.

This paper is organized as follows. Section II illustrates the principle of CRF. Section III presents the proposed framework of Co-CRF. The experimental results are showed in section IV and conclusions are made in section V.

II. Conditional Random Fields

A. Definition

CRF is an undirected graphical model that encodes a conditional probability distribution using a given set of features. In what follows, X is a input sequence to be labeled, and Y is a sequence of random variables corresponding to label sequences. Y is assumed to range over a finite label alphabet $\mathcal{Y}$. The random variables X and Y are jointly distributed, but in a discriminative framework we construct a conditional model $p(Y|X)$ from paired observation and label sequences. Formally, we define $G = (V, E)$ to be an undirected graph such that there is a node $v \in V$ corresponding to each of the random variables representing an element Y_v of Y. If each random variable Y_v obeys the Markov property with respect to the graph: $p(Y_v|X, Y_w, w \neq v) = p(Y_v|X, Y_w, w \sim v)$, where $w \sim v$ means that w and v are neighbors in G. Then (Y, X) is a conditional random field [6].

Simple chain or line is extensively used as G in the CRF model: $G = (V = 1, 2, ..., m, E = (i, i+1))$. And we will pay great attention to sequences $X = (X_1, X_2, ..., X_n)$ and $Y = (Y_1, Y_2, ..., Y_n)$ in this paper. The graph structure represented above is much important to structured prediction task. It avoids the information losing caused by learning just a per-word classifier. The linear conditional random field could be represented as a log-linear model:

$$p(y|x; w) = \frac{1}{Z(x, w)} \exp \sum_j w_j F_j(x, y). \quad (1)$$

We assume that each feature-function F_j is actually a sum along the sequence, for $i = 1$ to $|\mathcal{Y}|$:

$$F_j(x, y) = \sum_i f_j(y_{i-1}, y_i, x, i). \quad (2)$$

This means that the up-level feature-function F_j is computed based on the low-level feature-functions f_i, and the low-level feature-function can depend on the whole processing unit, the current label, the previous label, and the current position i within the processing unit(Figure 1). And CRF uses an observation-dependent normalization factor which is accumulated as:

$$Z(x, w) = \sum_{y'} \exp \sum_j w_j F_j(x, y'). \quad (3)$$

The main task of training a CRF is to find the weight vector w that gives the best possible prediction

$$y^* = \arg\max_y p(y|x; w) \quad (4)$$

for each training example x.

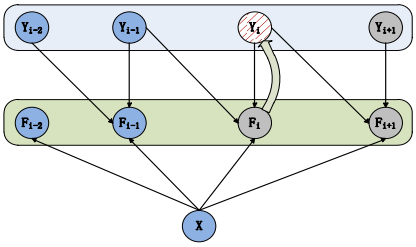

Figure 1. Feature-Function F_i in a chain-structured CRF. An black circle indicates that the variable is still unknown. Y_i is assigned by all the values in $\mathcal{Y}$.

B. Parameter Estimation for CRF

The essence of the CRF is finding the parameters of a statistical model that maximize the *conditional log-likelihood (CLL)* of the training data by *maximum-likelihood estimation (MLE)*. Gradient-descent method is much close to our proposal to maximize the CLL.

The stochastic gradient descent parameters are updated based on single training sample. Therefore, we evaluate the partial derivative of Equation 1 on a single training sample with respect to each w_j. First the log-likelihood is given by

$$\mathcal{L}(w) = \sum_s [\log \frac{1}{Z(x, w)} + \sum_j w_j F_j(x, y)] \quad (5)$$

The s in the equation indicates different samples. This function is concave [3], guaranteeing convergence to the global maximum.

Making partial derivative of the log-likelihood with respect to parameter w_j gives

$$\frac{\partial L(w)}{\partial w_j} = \sum_s [F_j(x, y) - \sum_{y'} F_j(x, y') p(y'|x; w)]. \quad (6)$$

And it could be written as

$$\frac{\partial \mathcal{L}(w)}{\partial w_j} = E_{y \sim \widetilde{p}(x,y)}[F_j(x, y)] - \sum_s E_{y' \sim p(y'|x;w)}[F_j(x, y')].$$

where $\widetilde{p}(x, y)$ is the empirical distribution of training data and $E_{y' \sim p(y'|x;w)}[\cdot]$ denotes expectation with respect to distribution p. So the partial derivation could be seen as the value of feature-function j for the true training label y, minus the mathematical expectation of the feature-function for all possible label y.

At the global maximum the sum of the gradients for each training sample in the entire training set T is zero, so we have

$$\sum_{(x,y) \in T} F_j(x, y) = \sum_{(x,\cdot)} E_{y \sim p(y|x;w)}[F_j(x, y)]. \quad (7)$$

The left side above is the sum value of feature-function j on the entire training set T. The right side is the total value of feature-function j predicted by the model.

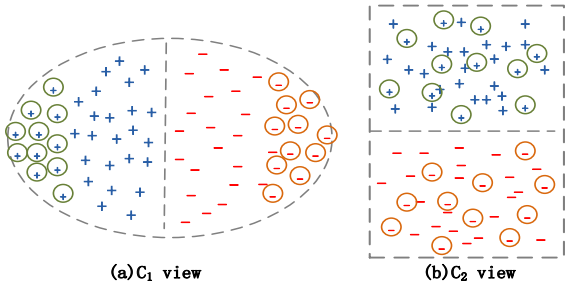

Figure 2. Conditional independent assumption on feature split. With this assumption the high confident data points in C_1 view, represented by circled labels, will be randomly scattered in C_2 view. This is advantageous if they are to be used to teach the classifier in C_2 view.

The parameters are updated based on the *improved iterative scaling (IIS)* algorithm [8]. And it can be represented as:

$$w_j := w_j + \alpha(F_j(x,y) - E_{y' \sim p(y'|x;w)}[F_j(x,y')]) \quad (8)$$

for each sample. And α in the equation is a learning rate parameter.

III. Conditional Random Fields with Co-training Framework

The previous section presents how to train a CRF classifier for sequence data. The accuracy of the CRF classifier is in proportion to the scale of training data set. Unfortunately the labeled training data are limited in real applications. To solve the contradiction between the accuracy and the training data, a semi-supervised learning framework named Co-training is introduced to the model.

Co-training splits the features of a sequence into two sets to train two separate CRF classifiers. And the two sub-feature sets are demanded to be conditional independent (Figure 2). More specifically, the splitting method is trying to weaken the linear independence between the sub-feature sets. Prim algorithm [2] is applied in the splitting procedure. We define $G' = (V', E')$ to be an undirected complete graph such that there is a node $v' \in V'$ corresponding to each of the features. The correlation coefficient among the features is the main linear correlation statistic. So it is regarded as the weight of each edges (E'). The splitting algorithm is introduced in Algorithm 1.

The two sub-feature sets conditional independently classify the sequence data, and each sub-feature set is sufficient to train a good classifier. So two separate CRF classifiers are trained on the two sub-feature sets respectively. And generally the training data are labeled teacher signals given by users. Then each CRF classifier tries to classify the unlabeled test data, and a few of the most confident samples are selected to 'teach' the other CRF classifier. Each CRF classifier is retrained with the new training samples given by the other CRF classifier, and the process repeats until k iterations. The pseudo code of this algorithm is provided in Algorithm 2.

Algorithm 1 The Splitting Method

Input: Weighted undirected complete graph with vertices V' and edges E'
Output: W and U represent the two sub-feature sets.

Find the minimal weight in E' to identify the two initial node$\{u, w\}$;
Initialize: $W = \{w\}$, $E_w = \{\}$, $U = \{u\}$, $E_u = \{\}$;
while $W \bigcup U <> V'$ **do**
  Choose an edge (w, x) with maximal weight such that w is in W and x is not in $W \bigcup U$;
  Add x to W, and (w, x) to E_w;
  Choose an edge (u, x) with maximal weight such that u is in U and x is not in $W \bigcup U$;
  Add x to U, and (u, x) to E_u;
end while

Algorithm 2 The Co-training Algorithm

Given:
1. Two sets $L_1 \& L_2$ of labeled training examples;
2. A set U of unlabeled test examples;
3. Two separated feature sets $F_1 \& F_2$;

Initialize the iteration parameter k, pool size u and leap size v;
for $i = 1$ to k **do**
  Create a temporary pool by randomly choosing u examples from U;
  Use L_1 to train CRF classifier C_1 using F_1 features;
  Use L_2 to train CRF classifier C_2 using F_2 features;
  Run C_1 to label u examples and select v examples it feels most confident V_1;
  Run C_2 to label u examples and select v examples it feels most confident V_2;
  Add V_1 examples to L_2;
  Add V_2 examples to L_1;
  Return the remaining examples back to U;
end for

Co-training assumes that each sub-feature set of the sequence data is sufficient to train a CRF classifier, so that the most confident samples selected by CRF classifier are credible to 'teach' the other CRF classifier. And the assumption that the sub-features are conditionally independent makes the selected data points are *idd* samples for the other CRF classifier. So the iteration procedure is beneficial to achieve a high accuracy with limited labeled training data. The Conditional Random Field with Co-training Framework is demonstrated in Figure 3.

Table I
Comparison between the proposed method and the baseline systems(Mean: the average accuracy in the classification process. Std: standard deviation of the average accuracy, which characterize the stability of the classification process). The best result in each column is indicated in boldface

		C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	Total
Mean	HMM	63.91	64.49	70.43	42.11	25.82	49.81	45.10	45.37	**79.12**	56.32	54.25
	CRF	**88.45**	45.40	72.48	51.25	58.89	58.87	**87.88**	**83.97**	48.41	68.72	66.63
	Co-CRF	86.97	**72.40**	**82.00**	**60.29**	**63.42**	**71.83**	83.99	69.04	70.45	**85.43**	**74.28**
Std	HMM	21.46	9.83	17.66	8.29	**6.28**	**4.06**	8.75	10.78	8.48	7.76	10.34
	CRF	6.90	9.76	**3.72**	**5.45**	9.88	10.68	6.78	**3.99**	9.11	12.35	2.40
	Co-CRF	**2.99**	**3.33**	3.81	7.30	12.60	7.36	**4.45**	4.95	**4.55**	**3.53**	**1.27**

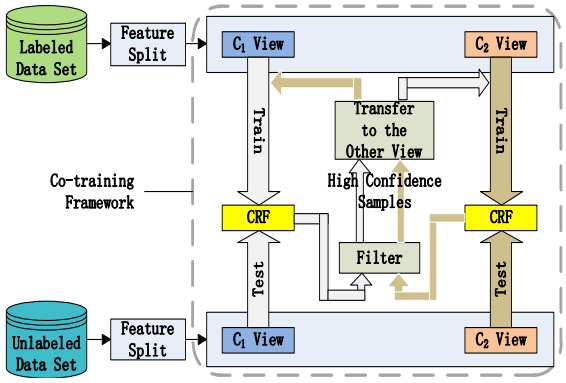

Figure 3. The Conditional Random Field with Co-training Framework.

IV. Experimental Results and Analysis

All the experiments are performed on the EMG physical action data set derived from UC Irvine machine learning repository* which consists of 4 separate subsets. And each subset includes 10 normal and 10 aggressive physical actions that measure the human activity, and we regard the different actions data as different classes to be classified. The overall number of features is 8, which corresponds to 8 input sequences one for a muscle channel. Each sequences contains 10000 samples.

We present three experiments to test the performance of the proposed method. First, we make comparisons between the proposed method and other baselines, arbitrarily choosing a set of fixed setting parameters including the number of co-training iterations k and the leap size v. We then test the trade-off between the co-training parameters(k & v) by fixing the total amount of unlabeled data, i.e., k*v sentences. In the last experiment, we test the impact of the iteration parameter k by fixing v to find the stable point of the iteration. Due to the randomness in our co-training algorithm, Every experiment repeats 10 times to report the average results.

A. Comparing with Baseline Systems

Table I shows the performance comparison of the proposed method and the baseline systems. Note that all the results shown in Table I are performed in different classifiers with a fixed number of labeled training data. And in the experiment 1000 labeled sequence elements are used to classify a unlabeled data set with 80000 sequence elements. It is necessary to announce that the Co-training parameters we used in the experiment are k=20 and v=20.

The empirical result shows that the Co-CRF outperforms the baseline systems with an accuracy of 74.28%. And the lowest standard deviation of the average accuracy presents that the Co-CRF performs a more stable result compared with the baseline systems. Theoretically, HMM ought to be the worst method in accuracy. However, HMM performs better than CRF and Co-CRF in a few of classes. Because HMM is affected by the Label Bias Problem, which makes it works instability. It is obvious that the standard deviations of CRF and Co-CRF are remarkable. So we summarize that overcoming the Label Bias Problem helps CRF classifier make a better decision. Furthermore, Co-training helps CRF to get a higher accuracy of the classification. Above all, the Co-CRF outperforms the baseline systems while the labeled training data is limited.

B. Trade-off between Co-training Parameters

The features of the data set are required to be split up in the Co-training algorithm. So the separate CRF classifiers of the Co-training system cannot be as good as the CRF classifier with all features, if there is no iteration step in the Co-training process. But the Co-training parameters can also make a great improvement to the accuracy of the CRF classification. This experiment is designed to test the trade-off between the iteration parameter k and the leap size v. The number of the unlabeled training data is fixed. And 400 unlabeled sentences (4000 unlabeled samples) are treated as the training data in the experiment. The unlabeled data is used to both test and train the CRF classifiers in the iteration process, while the number of the unlabeled data is still equal to k*v. Fig 4 shows the results. We see that when $k = 20$ and $v = 20$, the system's performance reaches a peak accuracy of 74.28%. In addition, accuracy increases 5.5% from $k = 2$ to $k = 10$, which suggests that our co-training algorithm needs at least 8-10 iterations to achieve satisfactory results. And we can give a summary that Co-CRF always offers a stable result of classification after a

*http://archive.ics.uci.edu/ml/datasets/EMG+Physical+Action+Data+Set

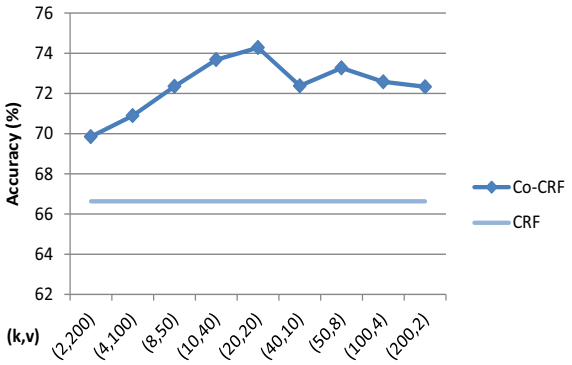

Figure 4. Trade-off between co-training parameters k and v with fixed amount of data

few numbers of iterations. And the detailed influence of the iteration parameter (k) is introduced in the next subsection.

C. Influence of the Iteration Parameter

In this experiment we try to measure the influence of the iteration parameter k in different leap size v. Several v values are fixed in the experiment. And we vary the number of iterations k to keep a record of the accuracy at each iteration. It is shown in Fig 5 that the accuracy of the Co-training grows with the increase of the iteration parameter k. However it is not a good option to increase k value extremely to get a high accuracy. Because the cost of system runtime grows exponentially as we increase the parameter k. In order to facilitate the time complexity comparison, we assume the time complexity of a training process with N samples to be $O(N)$. Note that the time consumed in testing or the other part of the experiment is ignored in time complexity comparison. For a separate CRF classifier, the training samples is increased by the iteration. The overall number of samples to train a CRF classifier is $Nk + \frac{k^2 v}{2}$. But there are two CRF classifiers to be trained in the Co-training structure. So we need to train $2Nk + k^2 v$ samples in the experiment. Finally, we get the time complexity to be $O(2Nk + k^2 v)$. It is obvious that the proposed method cannot offer a high accuracy if the iteration parameter k is not big enough. Whereas, it is time consuming to increase k value. Thus, there's a tradeoff between effectiveness and efficiency.

V. Conclusions

In this paper, we proposed a novel semi-supervised learning framework, Co-CRF, for high performance sequence data classification. The Co-CRF performs a structured prediction with taking the context into account. And it splits up the features into independent views, and adopts the proposed CRF strategy to select informative and representative feedback samples. Finally, the experimental results show that our Co-CRF method makes a considerable improvement compared with baseline algorithms. In this work, we focus on the co-training solutions for the classification problem

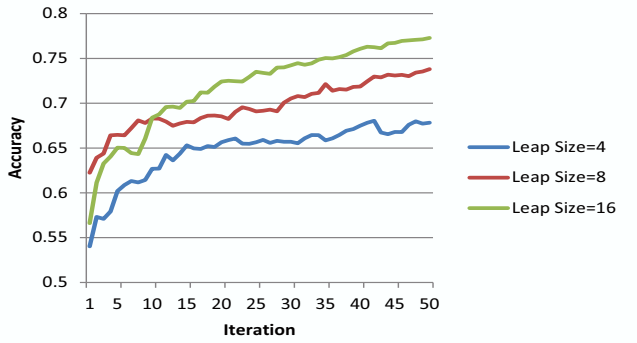

Figure 5. Performance varying the number of iterations k with fixed leap size v={4,8,16}

with Conditional Random Fields, and worth noting that the similar idea can be generally applied to other tasks (e.g. cluster) and models (e.g. HMM).

References

[1] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. *COLT'98 Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100, 1998.

[2] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. The algorithms of kruskal and prim. pages 631–638, 2009.

[3] C. Elkan. Log-linear models and conditional random fields. *ACM 17th Conference on Information and Knowledge Management(CIKM)*, 2008.

[4] Jaynes and E. T. Information theory and statistical mechanics. ii. *Physical Review*, 108:171–190, 1957.

[5] R. Klinger and K. Tomanek. Classical probabilistic models and conditional random fields. pages 15–23, December 2007.

[6] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *the Eighteenth International Conference on Machine Learning(ICML)*, pages 282–289, 2001.

[7] T. Mitchell. The role of unlabeled data in supervised learning. *Proceedings of the Sixth International Colloquium on Cognitive Science*, 1999.

[8] S. D. Pietra, V. D. Pietra, and J. Lafferty. Inducing features of random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 380–393, April 1997.

[9] Rabiner and L. R. A tutorial on hidden markov models and selected applications in speech recognition. *The IEEE77*, (2):257–286, 1989.

[10] S. Sarawagi and W. W. Cohen. Semi-markov conditional random fields for information extraction. *NIPS '04 Neural Information Processing Systems Foundation*, 2004.

[11] C. Sutton and A. McCallum. An introduction to conditional random fields for relational learning. pages 93–127, Novmber 2007.

[12] C. Sutton, K. Rohanimanesh, and A. McCallum. Dynamic conditional random fields: factorized probabilistic models for labeling and segmenting sequence data. *ICML '04 Proceedings of the twenty-first international conference on Machine learning*, page 99, 2004.