2D Conditional Random Fields for Web Information Extraction

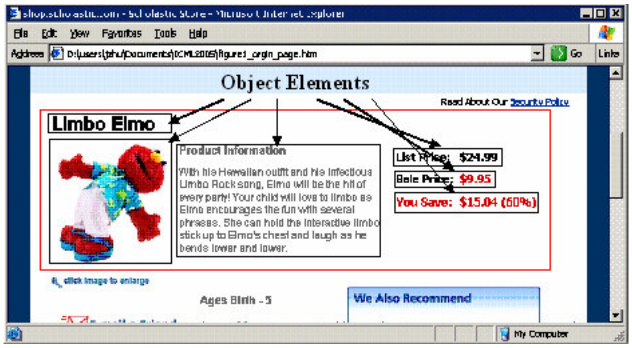
Jun Zhu[*][†][‡] JUN-ZHU@MAILS.TSINGHUA.EDU.CN
Zaiqing Nie[†] T-ZNIE@MICROSOFT.COM
Ji-Rong Wen[†] JRWEN@MICROSOFT.COM
Bo Zhang[‡] DCSZB@MAIL.TSINGHUA.EDU.CN
Wei-Ying Ma[†] WYMA@MICROSOFT.COM
[†]Microsoft Research Asia, Beijing, China
[‡]Department of Computer Science and Technology, Tsinghua University, Beijing, China

Abstract

The Web contains an abundance of useful semi-structured information about real world objects, and our empirical study shows that strong sequence characteristics exist for Web information about objects of the same type across different Web sites. Conditional Random Fields (CRFs) are the state of the art approaches taking the sequence characteristics to do better labeling. However, as the information on a Web page is two-dimensionally laid out, previous linear-chain CRFs have their limitations for Web information extraction. To better incorporate the two-dimensional neighborhood interactions, this paper presents a two-dimensional CRF model to automatically extract object information from the Web. We empirically compare the proposed model with existing linear-chain CRF models for product information extraction, and the results show the effectiveness of our model.

1. Introduction

While the Web is traditionally used for hypertext publishing and accessing, there are actually various kinds of objects embedded in static Web pages and dynamic Web pages generated from online Web databases. There is a great opportunity for us to extract and integrate all the related Web information about the same object together as an information unit, which is called a *Web object* (Nie et al., 2005). Typical Web objects are products, people, papers, organizations, etc. Commonly, objects of the same type obey the same structure or schema. We can imagine that once these objects are extracted and integrated from the Web, some large databases can be constructed to perform further knowledge discovery and data management tasks. This paper studies how to extend the existing information extraction techniques to automatically extract object information from Web pages.



With his Hawaiian outfit and his infectious Limbo Rock song, Elmo will be the hit of every party! Your child will love to limbo as Elmo encourages the fun with several phrases. She can hold the interactive limbo stick up to Elmo's chest and laugh as he bends lower and lower.

Figure 1. An object block with 6 elements (contained in the red rectangle) in a Web page

The information about an object in a Web page is usually grouped together as an *object block*, as shown in Figure 1. Using existing Web page segmentation (Cai et al., 2004) and data record extraction technologies (Liu et al., 2003; Zhai & Liu, 2005), we can automatically detect these object blocks, which are further segmented into atomic extraction entities called *object elements*. Each object element provides (partial) information about a single attribute of the Web object. For example, for the object block in Figure 1, the object elements provide information about the following attributes of a product object: Name, Image, Description, and Price. Given an object block with a set of elements, the Web object extraction task is to assign an attribute name to each object element.

Our empirical study shows that strong sequence characteristics exist for Web objects of the same type across different Web sites (see Section 5.1 for detailed discussion). Condition Random Fields (CRFs) (Lafferty et al., 2001) are the state of the art approaches in information extraction taking advantage of the sequence characteristics to do better labeling, compared with HMMs (Leek, 1997; Freitag & McCallum, 1999) and MEMMs (McCallum et al., 2000).

Appearing in *Proceedings of the 22nd International Conference on Machine Learning*, Bonn, Germany, 2005. Copyright 2005 by the author(s)/owner(s).

[*]The work is done when the author is visiting Microsoft Research Asia.

However, in order to use the existing linear-chain CRFs for Web object extraction, we have to first convert a two-dimensional object block (*i.e.* an object block whose elements are two-dimensionally laid out) into a sequence of object elements. Given the two-dimensional nature of object blocks, how to sequentialize them in a meaningful way could be very challenging. Moreover, as shown by our empirical evaluation, using the two-dimensional neighborhood dependencies (*i.e.* interactions between labels of an element and its the neighbors in both vertical and horizontal directions) in Web object extraction could significantly improve the extraction accuracy.

To better incorporate the two-dimensional neighborhood dependencies, a two-dimensional Conditional Random Field (*2D CRF*) model is proposed in this paper. We present the graphical representation of the 2D CRF model as a 2D grid (see Figure 2) and reformulate the conditional distribution by defining some matrix random variables. Then we deduce the forward-backward vectors based on the reformulated conditional distribution for efficient parameter estimation and labeling. Since the sizes of the elements in an object block can be arbitrary, we introduce the concept of *virtual states* to model an object block as a 2D grid. We compare our model with linear-chain CRF models for product information extraction and the experimental results show that our model significantly outperforms linear-chain CRF models in scenarios with two-dimensional neighborhood dependencies.

The rest of this paper is organized as follows. We discuss the related work in the next section. In section 3, 2D CRFs are presented, and the parameter estimation and labeling methods are discussed. Section 4 introduces a novel way of modeling an object block as a 2D grid. In section 5, we present our experimental setup and results. Section 6 brings this paper to a conclusion and finally we give our acknowledgements.

2. Related Work

To the best of our knowledge, we haven't seen any work on incorporating the two-dimensional neighborhood interactions into a graphical model for Web information extraction. However, there have been some previous attempts to incorporate complex interactions between labels.

Sutton et al., (2004) proposed Dynamic Conditional Random Fields. As a particular case, a factorial CRF (FCRF) was used to jointly solve two NLP tasks (noun phrase chunking and part-of-speech tagging) on the same observation sequence. Improved accuracy was achieved by modeling the interactions between the two tasks.

Bunescu and Mooney (2004) used a relational Markov network (Taskar et al., 2002) to collectively classify the entities in a document, achieving increased accuracy by learning dependencies between similar entities. Our work differs from theirs in two main aspects. First of all, Bunescu et al. focused on extracting named entities from natural language documents. It can not be directly applied to solve our problem. Since most of its global clique templates, such as overlap template and repeat template, are no longer useful in our scenario. Secondly, although approaches by Bunescu et al. (2004) and Taskar et al. (2002) can be extended to solve our problem by only defining neighbor-based global clique templates, the resulting graph and its factor graph always contain cycles. It's well-known that exact inference on graphs with cycles can be too expensive. Thus, Bunescu et al. used the Voted Perceptron (Collins, 2002b) algorithm to train their model and the max-product (Kschischang, et al., 2001) algorithm to do labeling. However, the Voted Perceptron algorithm has its limitations (Collins, 2002a). The max-product algorithm must take an iterative procedure to approximate the marginal distributions for a graph with cycles. In our approach, instead of doing inference on the graph directly, we present it on a 2D grid. By marginalizing variables progressively along the diagonals, we can use a more efficient gradient-based method to train our model and a more efficient dynamic programming method to do labeling. Please see Section 5.2.4 for an empirical comparison.

Previous work on 2D models can be found in Image Processing and Computer Vision. Li et al. (2000) proposed two-dimensional Hidden Markov Models (2D HMMs) for image classification. Since 2D HMMs are generative models, some conditional independence assumption is made for computational tractability. Markov Random Fields (Besag, 1974; Li, 2001) in image analysis are also generative models, but unlike 2D HMMs, MRFs model the prior distribution over labels as a markov random field. For computational tractability, the conditional independence assumption is also made as 2D HMMs. Recently, to take the advantages of conditional models (like CRFs), Discriminative Random Fields (DRFs) were proposed by Kumar et al. (2003; 2004) in the case of binary image classification. Based on CRFs, DRFs model the association potential as well as the interactions between the neighboring sites on a 2D grid. It has been shown that DRFs outperform MRFs in natural image classification. However, the proposed DRFs are limited to binary classification, so they can't be applied to our application.

Liu et al. (2003) and Zhai & Liu (2005) proposed an unsupervised approach to automatically detect Web blocks and extract the Web data from the blocks, which can be used as the input of our model for labeling the extracted data. Since we are focusing on assigning the semantic label to their extracted data, our work can be seen as complementary to theirs.

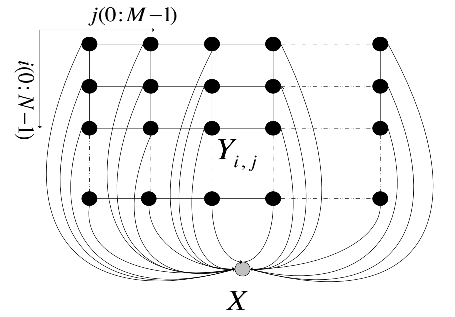


Figure 2. The graphical structure of 2D CRFs

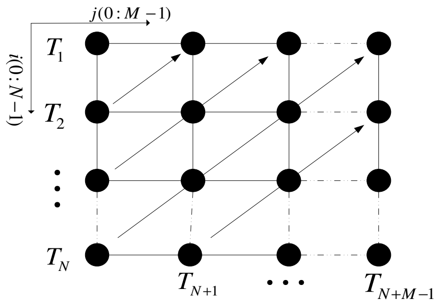


Figure 3. The diagonal state sequences of a 2D grid

3. 2D Conditional Random Fields

In this section we first introduce the basic concepts of Conditional Random Fields and present the conditional distribution of linear-chain CRFs (a particular case of CRFs), and then we discuss the conditional probability over labels of 2D CRFs, and finally we discuss how to estimate the parameters and how to perform labeling.

3.1 Linear-chain CRFs

Conditional random fields (Lafferty et al., 2001) are undirected graphical models. As defined before, X is a random variable over the observations to be labeled, and Y is a random variable over corresponding labels. All components Y_i of Y are assumed to range over a finite label alphabet $\mathcal{Y}$. CRFs construct a conditional model $p(Y \mid X)$ with a given set of features from paired observations and labels.

CRF Definition:

Let $G=(V,E)$ *be an undirected graph such that* $Y=\{Y_v\}_{v\in V}$. *Then* (X,Y) *is said to be a conditional random field if, when conditioned on* X, *the random variables* Y_v *obey the Markov property with respect to the graph:* $p(Y_v \mid X, Y_{V-\{v\}}) = p(Y_v \mid X, Y_{N_v})$, *where* $V-\{v\}$ *is the set of nodes in the graph except the node* v *and* N_v *is the set of neighbors of the node* v *in graph* G.

Thus, a CRF is a random field globally conditioned on the observations X. Linear-chain CRFs were first introduced by Lafferty et al. (2001). By the Hammersley-Clifford Theorem (Hammersley & Clifford, 1971), the conditional distribution of the labels y given the observations x has the form,

$$p(\mathrm{y} \mid \mathrm{x}) = \frac{1}{Z(\mathrm{x})} \exp\left(\sum_{e\in E,k} \lambda_k f_k(e, \mathrm{y}|_e, \mathrm{x}) + \sum_{v\in V,k} \mu_k g_k(v, \mathrm{y}|_v, \mathrm{x}) \right)$$

where $\mathrm{y}|_e$ and $\mathrm{y}|_v$ are the components of y associated with edge e and vertex v respectively; f_k and g_k are feature functions; λ_k and μ_k are parameters to be estimated from the training data and $Z(\mathrm{x})$ is the normalization factor, also known as partition function, which has the form,

$$Z(\mathrm{x}) = \sum_{\mathrm{y}} \exp\left(\sum_{e\in E,k} \lambda_k f_k(e, \mathrm{y}|_e, \mathrm{x}) + \sum_{v\in V,k} \mu_k g_k(v, \mathrm{y}|_v, \mathrm{x}) \right)$$

3.2 2D CRFs

As we mentioned before, linear-chain CRFs cannot represent two-dimensional neighborhood interactions. In this section, we introduce a two-dimensional CRF (2D CRF) model which is also a particular case of CRFs. The graphical structure of 2D CRFs is a 2D grid (see Figure 2). Here we use X to denote the random variable over observations, and Y to denote the random variable over the corresponding labels. $Y_{i,j}$ is a component of Y at the vertex (i, j). The cliques of this graph are its edges and vertices, so the conditional distribution has the *same* form as linear-chain CRFs. 2D CRFs can also be viewed as a finite-state model. Each variable $Y_{i,j}$ has a finite set of state values and we assume the one-to-one mapping between states and labels. In this paper, we denote the label and the state value at (i, j) using the same notation $\mathrm{y}_{i,j}$.

In the following, we first define some matrix random variables along the diagonals of the model's graph (see Figure 3) and reformulate the conditional distribution. Then we deduce the forward-backward vectors (see Section 3.3 & 3.4) for efficient training and labeling. We first introduce some notations we use as follows:

1) The state sequence on diagonal d $(1 \le d \le M+N-1)$, $\{\mathrm{y}_{d-1,0}, \mathrm{y}_{d-2,1}, \cdots, \mathrm{y}_{0,d-1}\}$ is denoted by T_d.
2) Two special state sequences are added: $T_0 = start$ and $T_{N+M} = stop$.
3) The diagonal on which the random variable $Y_{i,j}$ lies is denoted by $\Delta(i, j)$.
4) The set of coordinates of the random variables on diagonal d, $\{(i, j), \Delta(i, j) = d\}$ is denoted by $I(d)$.
5) The set of edges between diagonals $d-1$ and d $\{((i', j'), (i, j)) \in E : (i', j') \in I(d-1) and\ (i, j) \in I(d)\}$ is denoted by $E(d)$.

For each diagonal d, we define the $|\mathcal{Y}|^{S_{d-1}} \times |\mathcal{Y}|^{S_d}$ matrix random variable $M_d(\mathrm{x}) = [M_d(T'_{d-1}, T_d \mid \mathrm{x})]$ by

$$M_d(T'_{d-1}, T_d \mid \mathrm{x}) = \exp(\Lambda_d(T'_{d-1}, T_d \mid \mathrm{x}))$$

$$\Lambda_d(T'_{d-1}, T_d \mid \mathrm{x}) = \sum_{e\in E(d),k} \lambda_k f_k(e, \mathrm{y}'_{i',j'}, \mathrm{y}_{i,j}, \mathrm{x}) + \sum_{v\in I(d),k} \mu_k g_k(v, \mathrm{y}_{i,j}, \mathrm{x}) \tag{1}$$

where $T'_{d-1}=\{\mathrm{y}'_{d-2,0},\mathrm{y}'_{d-3,1},\cdots,\mathrm{y}'_{0,d-2}\}$, $T_d=\{\mathrm{y}_{d-1,0},\mathrm{y}_{d-2,1},\cdots,\mathrm{y}_{0,d-1}\}$; S_{d-1} and S_d are the number of states in T'_{d-1} and T_d respectively; $e=((i',j'),(i,j))$ and $v=(i,j)$.

Thus, when given the observations x and the parameters, the matrices can be computed as needed. The normalization factor $Z(\mathrm{x})$ can be expressed as the $(start, stop)$ entry of the product of these matrices:

$$Z(\mathrm{x})=\left(M_1(\mathrm{x})M_2(\mathrm{x})\cdots M_{M+N}(\mathrm{x})\right)_{(start,stop)}$$

For a particular combination of label assignments y given the observations x , let $T_d=\{\mathrm{y}_{d-1,0},\mathrm{y}_{d-2,1},\cdots,\mathrm{y}_{0,d-1}\}$ $(1\le d\le M+N-1)$, and then the conditional probability has the form,

$$p(\mathrm{y}\,|\,\mathrm{x})=\frac{\prod_{d=1}^{M+N} M_d(T_{d-1},T_d\,|\,\mathrm{x})}{\left(\prod_{d=1}^{M+N} M_d(\mathrm{x})\right)_{(start,stop)}} \qquad (2)$$

where $T_0=start$ and $T_{N+M}=stop$.

3.3 Parameter Estimation

Given the training data $D=\{(\mathrm{y}^i,\mathrm{x}^i)\}_{i=1}^N$ with the empirical distribution $\tilde{p}(\mathrm{x},\mathrm{y})$, the log-likelihood of $\tilde{p}(\mathrm{x},\mathrm{y})$ with respect to a conditional model $p(\mathrm{y}\,|\,\mathrm{x},\Theta)$ is defined as,

$$L(\Theta)=\prod_{\mathrm{x},\mathrm{y}}\tilde{p}(\mathrm{x},\mathrm{y})\log p(\mathrm{y}\,|\,\mathrm{x},\Theta)$$

The parameter estimation problem is to find a set of parameters $\Theta=\{\lambda_1,\lambda_2,\cdots;\mu_1,\mu_2,\cdots\}$ that optimize the concave log-likelihood function. The function can be optimized by the techniques used in other maximum-entropy models (Lafferty et al., 2001; Berger et al., 1996). We choose gradient-based L-BFGS (Liu & Nocedal, 1989) for its outstanding performance over other optimization algorithms for linear-chain CRFs (Malouf, 2002; Sha et al., 2003). Each element of the gradient vector is given by

$$\frac{\partial L(\Theta)}{\partial\lambda_k}=E_{\tilde{p}(\mathrm{x},\mathrm{y})}[f_k]-E_{p(\mathrm{y}|\mathrm{x},\Theta)}[f_k]$$

where $E_{\tilde{p}(\mathrm{x},\mathrm{y})}[f_k]$ is the expectation with respect to the empirical distribution and can be easily computed and $E_{p(\mathrm{y}|\mathrm{x},\Theta)}[f_k]$ is the expectation with respect to the conditional model distribution. For feature function f_k ,

$$E_{p(\mathrm{y}|\mathrm{x},\Theta)}[f_k]=\sum_{\mathrm{x}}\tilde{p}(\mathrm{x})\sum_{e\in E}\sum_{\mathrm{y}'_{i',j'},\mathrm{y}_{i,j}} p(\mathrm{y}'_{i',j'},\mathrm{y}_{i,j}\,|\,\mathrm{x})f_k(e,\mathrm{y}'_{i',j'},\mathrm{y}_{i,j},\mathrm{x})$$

where $e=((i',j'),(i,j))$, and for feature function g_k ,

$$E_{p(\mathrm{y}|\mathrm{x},\Theta)}[g_k]=\sum_{\mathrm{x}}\tilde{p}(\mathrm{x})\sum_{v=(i,j)\in V}\sum_{\mathrm{y}_{i,j}} p(\mathrm{y}_{i,j}\,|\,\mathrm{x})g_k(v,\mathrm{y}_{i,j},\mathrm{x})$$

So the main computation is to compute the marginal probabilities, which are needed to compute the gradients at each iteration. The idea of forward-backward algorithm can be extended here to reduce computation. As the conditional distribution has the form in equation (2), the state sequence T_d is in fact an "isolating" element in the expansion of $p(\mathrm{y}\,|\,\mathrm{x})$, which plays the same role as a state at a single unit of time in linear-chain CRFs.

For each diagonal index $d=0,\cdots,M+N$, the forward vectors $\alpha_d(\mathrm{x})$ are defined with base case

$$\alpha_0(T_0\,|\,\mathrm{x})=\begin{cases}1 & if\ T_0=start\\ 0 & otherwise\end{cases}$$

and with recurrence $\alpha_d(\mathrm{x})=\alpha_{d-1}(\mathrm{x})M_d(\mathrm{x})$

Similarly, the backward vectors $\beta_d(\mathrm{x})$ are defined as

$$\beta_{N+M}(T_{M+N}\,|\,\mathrm{x})=\begin{cases}1 & if\ T_{M+N}=stop\\ 0 & otherwise\end{cases}$$

and $\qquad \beta_d(\mathrm{x})^{\mathrm{T}}=M_{d+1}(\mathrm{x})\beta_{d+1}(\mathrm{x})$

Thus, the marginal probability of being in state sequence T_d on diagonal d given the observations x is

$$p(T_d\,|\,\mathrm{x})=\frac{\alpha_d(T_d\,|\,\mathrm{x})\beta_d(T_d\,|\,\mathrm{x})}{Z(\mathrm{x})}$$

So the marginal probability of being at state $\mathrm{y}_{i,j}$ at $Y_{i,j}$ on diagonal d is $\quad p(\mathrm{y}_{i,j}\,|\,\mathrm{x})=\sum_{T_d:T_d(i,j)=\mathrm{y}_{i,j}} p(T_d\,|\,\mathrm{x})$

Similarly, the marginal probability of being in state sequence T'_{d-1} on diagonal $d-1$ and T_d on diagonal d is

$$p(T'_{d-1},T_d\,|\,\mathrm{x})=\frac{\alpha_{d-1}(T'_{d-1}\,|\,\mathrm{x})M_d(T'_{d-1},T_d\,|\,\mathrm{x})\beta_d(T_d\,|\,\mathrm{x})}{Z(\mathrm{x})}$$

So the marginal probability of being at state $\mathrm{y}'_{i',j'}$ at $Y_{i',j'}$ and $\mathrm{y}_{i,j}$ at $Y_{i,j}$ is,

$$p(\mathrm{y}'_{i',j'},\mathrm{y}_{i,j}\,|\,\mathrm{x})=\sum_{T'_{d-1}:T'_{d-1}(i',j')=\mathrm{y}'_{i',j'}}\ \sum_{T_d:T_d(i,j)=\mathrm{y}_{i,j}} p(T'_{d-1},T_d\,|\,\mathrm{x})$$

where $((i',j'),(i,j))\in E(d)$.

3.4 Labeling

Labeling is the task to find labels y^* that best describe the observations x , that is, $\mathrm{y}^*=\max_{\mathrm{y}} p(\mathrm{y}\,|\,\mathrm{x})$

Dynamic programming algorithms are the most desirable methods for this problem. For 2D HMMs, Li et al. (2000) have proposed the variable-state Viterbi algorithm and the difference from the normal Viterbi algorithm is that the number of possible state sequences at every position in the viterbi transition diagram is exponential to the number of states on the diagonal. For 2D CRFs, the variable-state Viterbi algorithm can be used directly for the "isolating" element T_d. However, as the dimensions of the transition matrices $M_d(\mathrm{x})$ are exponential to the state numbers in T'_{d-1} and T_d respectively, the computational complexity can also be very high. To reduce computation, Li et al. (2000) propose a path-constrained suboptimal method. Li et al. choose N most likely state sequences out of all the state sequences based on the assumption that the random variables on a diagonal are statistically independent when the diagonal is separated from others. Based on the same independence assumption, we can use the path-constrained suboptimal method to compute the approximate gradients for the L-BFGS (Liu et al., 1989) algorithm to train our model, and variable-state Viterbi (Li et al., 2000) algorithm to find the best state sequence.

4. Modeling an Object Block

As described in the introduction section, an object block is composed of some atomic extracted entries called object elements. To use the proposed 2D CRFs for Web information extraction, we first index the object elements on a 2D grid according to their position and size information. The two-dimensionally indexed object block of Figure 1 is shown in Figure 4, where we have used $\mathrm{x}_{i,j}$ to denote the element at (i, j) and x_{null} to denote the null elements (*i.e.* the elements that don't exist). Then we need to associate each element $\mathrm{x}_{i,j}$ with a state $\mathrm{y}_{i,j}$ and null element x_{null} with null state y_{null} (*i.e.* the state that doesn't exist). At the second step we need to handle the irregular neighborhood dependencies caused by the elements' arbitrary sizes in a Web page. Take the block in Figure 4 as an example, the element $\mathrm{x}_{1,1}$ is so large that elements $\mathrm{x}_{1,2}$, $\mathrm{x}_{2,2}$ and $\mathrm{x}_{3,2}$ are all its neighbors. But for our 2D model, if we associate each element with only one state as in Figure 5(a), the association result can not represent the neighborhood dependencies between state pairs $(\mathrm{y}_{1,1}, \mathrm{y}_{2,2})$ and $(\mathrm{y}_{1,1}, \mathrm{y}_{3,2})$. To model these neighborhood dependencies, we introduce some *virtual states* to avoid further segmenting the atomic extracted object elements into smaller ones. We denote the states associated with object elements as *real states*, and *virtual states* are mirrors of the real states. The virtual states and the corresponding real states must have the same values when a transition takes place. We will use $\mathrm{y}^{*}_{i,j}$ to denote the virtual state of the real state $\mathrm{y}_{i,j}$.

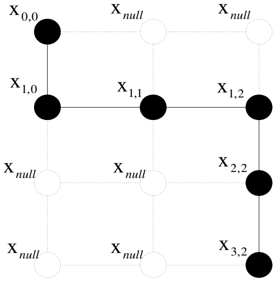


Figure 4. The two-dimensionally indexed object block

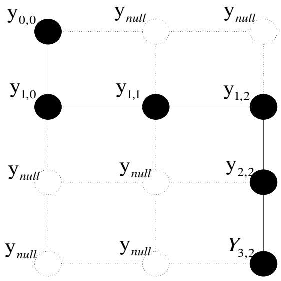


Figure 5(a). One-one association

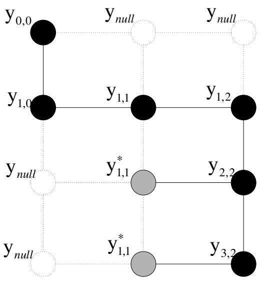


Figure 5(b). Association with virtual states

For each element, we define four neighbors (left, top, right and bottom) as the neighbors of the state with which it is associated. We denote the four neighbors of element $\mathrm{x}_{i,j}$ by a quad-tuple $(l_{i,j}, t_{i,j}, r_{i,j}, b_{i,j})$, where $l_{i,j}$, $t_{i,j}$, $r_{i,j}$ and $b_{i,j}$ are the coordinates of the left, top, right and bottom neighbors respectively or *null* if the corresponding neighbors don't exist. The neighbors are determined as follows: if $\mathrm{x}_{i,j}$ has only one left, top, right, or bottom neighbor, the corresponding neighbor is that one; If $\mathrm{x}_{i,j}$ has more than one left or right neighbors, the left or right neighbor is the highest one; If $\mathrm{x}_{i,j}$ has more than one top or bottom neighbors, the top or bottom neighbor is the most left one. Thus, the neighbors of each element in Figure 4 are:

$$\begin{aligned}
\mathrm{x}_{0,0} &: \left(null, null, null, (1,0)\right) \\
\mathrm{x}_{1,0} &: \left(null, (0,0), (1,1), null\right) \\
\mathrm{x}_{1,1} &: \left((1,0), null, (1,2), null\right) \\
\mathrm{x}_{1,2} &: \left((1,1), null, null, (2,2)\right) \\
\mathrm{x}_{2,2} &: \left((1,1), (1,2), null, (3,2)\right) \\
\mathrm{x}_{3,2} &: \left((1,1), (2,2), null, null\right)
\end{aligned}$$

The association result with virtual states is shown in Figure 5(b). Since the null states are ignored during inference, a diagonal state sequence is composed of the real and virtual states on that diagonal. Thus, the diagonal state sequences in Figure 5(b) are:

$$\begin{aligned}
T_1 &: \{\mathrm{y}_{0,0}\} \\
T_2 &: \{\mathrm{y}_{1,0}\} \\
T_3 &: \{\mathrm{y}_{1,1}\} \\
T_4 &: \{\mathrm{y}^{*}_{1,1}, \mathrm{y}_{1,2}\} \\
T_5 &: \{\mathrm{y}^{*}_{1,1}, \mathrm{y}_{2,2}\} \\
T_6 &: \{\mathrm{y}_{3,2}\}
\end{aligned}$$

We define an edge as a *virtual edge* if the edge's one end is associated with a virtual state and the other end is associated with the same real state, or the edge's two ends are both associated with the same virtual state. We define the other edges as real edges if the edges are not associated with null states. In Figure 5(b), real edges are solid and virtual edges are dotted. The virtual edges are not taken into account for the probability distribution, but they constrain the two states associated with them to have the same state value when a transition takes place. Thus, equation (1) is reformulated as:

$$\Lambda_d\left(T'_{d-1}, T_d \mid \mathrm{x}\right) = \begin{cases} -\infty, & \exists\,((i',j'),(i,j)) \in E_v(d) \;\; s.t.\; \mathrm{y}'_{i',j'} \neq \mathrm{y}_{i,j} \\ \sum\limits_{e \in E_r(d),k} \lambda_k f_k(e, \mathrm{y}'_{i',j'}, \mathrm{y}_{i,j}, \mathrm{x}) + \sum\limits_{v \in I_r(d),k} \mu_k g_k(v, \mathrm{y}_{i,j}, \mathrm{x}), & \text{otherwise} \end{cases}$$

where $E_v(d)$ and $E_r(d)$ are the sets of virtual edges and real edges between diagonals $d-1$ and d respectively; $I_r(d)$ is the set of coordinates of the real states on diagonal d and $e=((i^{'},j^{'}),(i,j))$, $v=(i,j)$.

5. Experimental Studies

In this section, we first conduct some statistical work to demonstrate that strong sequence characteristics exist for Web objects of the same type across different Web sites. Then we compare our model with linear-chain CRFs in the domain of product information extraction. We also present some empirical studies to compare Bunescu et al. (2004)'s work and our approach.

5.1 Statistical Results

To demonstrate that strong sequence characteristics exist for Web objects of the same type across different Web sites, we conduct our statistical study on two types of Web objects, products and researchers' homepages. We randomly collect 100 product pages (964 product blocks) and 120 homepages from different Web sites. For product objects, the attributes "name", "image", "price" and "description" are surveyed and for researchers' homepages, the attributes "name", "telephone", "email" and "address" are considered. We decide the sequence order of the elements in a web page in a top-down and left-right manner based on their position information. Basically, the element in the top level will be ahead of all the elements below it and for the elements at the same level, the left elements will be ahead of their right elements.

Table 1. Statistical results for objects from both product pages and homepages. We have used "DESC" instead of "DESCRIPTION" and "TEL" instead of "TELEPHONE" for space.

PRODUCT	BEFORE	HOMEPAGE	BEFORE
(NAME, DESC)	1.000	(NAME, TEL)	1.000
(NAME, PRICE)	0.987	(NAME, EMAIL)	1.000
(IMAGE, NAME)	0.941	(NAME, ADDRESS)	1.000
(IMAGE, PRICE)	0.964	(ADDRESS, EMAIL)	0.847
(IMAGE, DESC)	0.977	(ADDRESS, TEL)	0.906

The statistical results show that strong sequence characteristics exist among most attribute pairs in both types of objects. For example, a product's name is always ahead of its description in all the pages.

5.2 Performance Evaluation

We carry out our experiments in the domain of product object information extraction because of its plentiful spatial information. In the experiments, four attributes ("name", "image", "price", and "description") are evaluated.

5.2.1 Datasets

We setup our datasets with 572 randomly crawled product Web pages, and use the Web page segmentation technology (Cai et al., 2004) to collect all the blocks containing product information. The block elements are detected using this technology at a finer granularity. An appropriate segmentation granularity is important when segmenting the elements, because either over-segmentation or less-segmentation will affect the extraction accuracy. In our experiments, we prefer over-segmentation, that is, we always prefer smaller elements when there is segmentation uncertainty.

We totally collect 2500 product blocks from the Web pages and all the blocks are manually labeled. Actually, there are two types of these blocks. In the first type, the elements in a block do not have two-dimensional neighborhood dependencies. So, for this type of data, our model actually performs like linear-chain CRFs. This data set is denoted by **ODS**. In the second type, the elements in a block do have two-dimensional dependencies. This type of dataset is denoted by **TDS**. 400 blocks from **TDS** and 100 blocks from **ODS** are randomly selected as training samples. The remaining datasets, each containing 1000 blocks, are our testing sets.

5.2.2 Model Construction and Training

To test our model's effectiveness of incorporating two-dimensional neighborhood interactions for Web IE, we choose linear-chain CRFs as the baseline models for their outstanding performance over other sequential models. Both the linear-chain CRFs and 2D CRFs are constructed with the same set of feature functions. In our experiments, we use an html-parser to accurately get all the used features, such as contents, font size, link-URL, image's source URL, etc. The two most important visual features in our approach are elements' **positions** and **sizes** which are used to determine the neighborhood dependencies. We approximate an element's size by a quadrangle's area. The quadrangle's height and width can be accurately got with the same parser.

Both models are trained on the same training data set using L-BFGS (Liu et al., 1989) algorithm. For the linear-chain CRF model, the gradients are exactly computed, and for our 2D CRF model, the gradients are approximately computed using the suboptimal method (Li et al., 2000) with the path number N set at 20. We use the convergence criterion,

$$\frac{\left|L\left(\Theta^{(k)}\right)-L\left(\Theta^{(k-1)}\right)\right|}{L\left(\Theta^{(k)}\right)}<\varepsilon$$

where, the relative tolerance is set $\varepsilon=10^{-7}$ as Malouf (2002). With 500 training samples and the same initial parameters 0.1, the linear-chain CRF model converges at 12 iterations, and the 2D CRF model at 13 iterations.

5.2.3 Experimental results and Analysis

We compare our model with the linear-chain CRF model on the testing sets **ODS** and **TDS**. The performance on each attribute is evaluated by *precision*, the percentage of returned elements that are correct; *recall*, the percentage of correct elements that are returned; and their harmonic mean *F1*. We also define two comprehensive evaluation criteria: (1) *block instance accuracy (BLK_IA)* as the percentage of blocks of which the key attributes (name, image, and price) are all correctly labeled, (2) *average F1 (AVG_F1)* as the average of *F1* values of different attributes.

Table 2. Evaluation results on the datasets **TDS** and **ODS**. We have used "DESC" in stead of "DESCRIPTION". 2D stands for 2D CRF model and LINEAR stands for linear-chain CRFs.

		TDS		ODS	
		2D	**LINEAR**	**2D**	**LINEAR**
PRECISION	NAME	0.911	0.790	0.794	0.762
	IMAGE	0.963	0.917	0.993	0.979
	PRICE	0.969	0.932	0.977	0.952
	DESC	0.849	0.828	0.772	0.813
RECALL	NAME	0.883	0.762	0.767	0.734
	IMAGE	0.963	0.917	0.993	0.979
	PRICE	0.919	0.895	0.942	0.945
	DESC	0.803	0.669	0.792	0.816
F1	NAME	0.897	0.776	0.781	0.745
	IMAGE	0.963	0.917	0.993	0.979
	PRICE	0.944	0.913	0.959	0.949
	DESC	0.824	0.740	0.782	0.814
AVG_F1		0.907	0.837	0.879	0.872
BLK_IA		0.756	0.600	0.782	0.755

The experimental results (table 2) show that for the dataset **ODS**, there is no significant difference between the linear-chain CRF model and our 2D CRF model because of data's sequential properties. But for dataset **TDS**, the performances on different attributes using our 2D CRF model are significantly improved. The precision and recall of the attribute "name" are both improved by 12%. Although the precision of the attribute "description" is improved only by 2.1%, the recall is improved by 13.4%. For attributes "image" and "price", a little smaller improvement is achieved, because these two attributes have notable state-information. For example, prices must contain formatted numbers, and images are all with empty texts and nonempty image source URLs. Thus, only using this information can give good results, and the neighborhood dependencies, which are represented by the transition feature functions in the models, don't contribute too much. From the average *F1*, we can also see the contribution of the neighborhood dependencies to improve the extraction accuracy. For one dimensional dataset **ODS**, the improvement of AVG_F1 is neglectable; but for **TDS**, the improvement is 7.7%. The block instance accuracy says the same thing: the improvement of using 2D CRFs on the dataset **TDS** is 15.6%, but the improvement on the dataset **ODS** is just 2.7%. Thus, the 2D CRF model significantly outperforms linear-chain CRF models for the two-dimensional Web information extraction.

5.2.4 COMPARED WITH VOTED PERCEPTRON ALGORITHM

As we mentioned before, by marginalizing variables progressively along the diagonals, we can use a more efficient gradient-based method L-BFGS to train our model, while Bunescu et al. used the Voted Perceptron (Collins, 2002b) algorithm to train their model because the computation of the gradient vectors is prohibitively expensive in their approach.

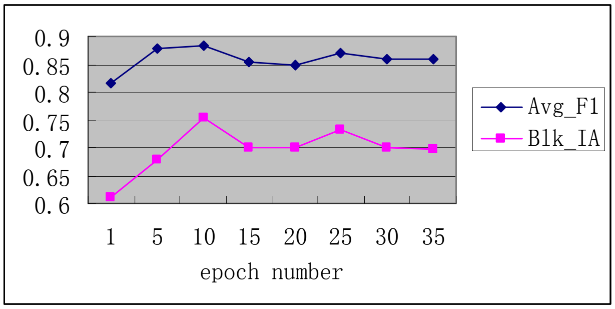


Figure 6. Evaluation results of Voted Perceptron algorithm

To show the effectiveness of our approach against theirs, we use the Voted Perceptron (Bunescu et al., 2004) algorithm to train our two-dimensional CRF model. The performance is evaluated on the testing set **TDS**, and the results are shown in Figure 6. We set the learning rate at 0.01 which was used by Bunescu et al. (2004). A series of epoch numbers are chosen as in Figure 6. From the results, we can see that when the epoch number is set at 10, the Avg_F1 (average F1) and Blk_IA (block instance accuracy) both achieve their highs, with Avg_F1 **0.8826** and Blk_IA **0.754**. Since there is no systematic way of selecting the best epoch number, it's possible that their approach performs much worse than the best performance which is still inferior to our 2D CRF model with L-BFGS training its parameters. Moreover, Collins (2002a) has pointed out that the Voted Perceptron algorithm's performance guarantees depend on the notion of "separability" of training samples. If the training samples are not separable, the algorithm will not converge. In our experiments, the Voted Perceptron algorithm doesn't converge within **3500** iterations, but the L-BFGS algorithm converges rapidly (within 13 iterations).

6. Conclusions

In this paper, we propose a two-dimensional Conditional Random Field model. This model provides a novel way of incorporating two-dimensional neighborhood dependencies to improve the performance of Web information extraction. By marginalizing variables progressively along the diagonals, efficient parameter learning and labeling can be performed. When the proposed model is applied to product information extraction, significant improvements are achieved compared with linear-chain CRF models.

Acknowledgements

We thank the anonymous reviewers for many detailed and valuable comments and Xin Zheng for many helpful discussions. The authors Jun Zhu and Bo Zhang are partially supported by National Natural Science Foundation of China (60135010, 60321002), and Chinese National Key Foundation Research and Development Plan (2004CB318108).

References

Besag, J. (1974). Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society, Series B,* 36:192-236.

Berger, A. L., Pietra, S. A. D., & Pietra, V. J. D. (1996). A maximum entropy approach to natural language processing. *Computational Linguistics*, 22, 39-71.

Bunescu, R. C., & Mooney, R. J. (2004). Collective information extraction with relational Markov networks. *In Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pp. 439-446, Barcelona, Spain.

Cai, D., Yu, S., Wen, J. R., & Ma, W. Y. (2004). Block-based web search. *In ACM SIGIR Conference*, 2004.

Collins, M. (2002a). Discriminative Training Methods for Hidden Markov Models: Theory and Experiments with Perceptron Algorithm. In *Proceedings of EMNLP*, 2002.

Collins, M. (2002b). Ranking algorithms for named-entity extraction: Boosting and the voted perceptron. *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL-02)* (pp. 489-496). Philadelphia, PA.

Freitag, D. & McCallum, A. (1999). Information Extraction with HMMs and shrinkage. In *Proceedings of the AAAI-99 Workshop on Machine Learning for Information Extraction*.

Hammersley, J., & Clifford, P. (1971). Markov fields on finite graphs and lattices. Unpublished manuscript.

Kschischang, F. R., Frey, B. J., & Loeliger, H.-A. (2001). Factor Graphs and the Sum-Product Algorithm. *IEEE Transactions on Information Theory*, 47(2):498-519.

Kumar, S., & Hebert, M. (2003). Discriminative random fields: A discriminative framework for contextual interaction in classification. *IEEE Int, Conf. on Computer Vision*, 2:1150-1157.

Kumar, S., & Hebert, M. (2004). Discriminative Fields for Modeling Spatial Dependencies in Natural Images. *Advances in Neural Information Processing Systems, NIPS 16*.

Lafferty, J., McCallum, A., & Pereira, F. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *In Proc. ICML.*

Leek, T. (1997). Information extraction using hidden Markov models. Master's thesis, University of California, San Diego.

Li, J., Najmi, A., & Gray, R. M. (2000). Image Classification by a Two-dimensional Hidden Markov Model. *IEEE Trans on Signal Processing,* Vol. 48, No. 2.

Li, S. Z. (2001). Markov Random Field Modeling in Image Analysis, Springer-Verlag, Tokyo.

Liu, B., Grossman, R. & Zhai, Y. (2003). Mining data records in web pages. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Liu, D. C., & Nocedal, J. (1989). On The Limited Memory BFGS Method for Large Scale Optimization. *Mathmetical Programming* 45, pp. 503-528.

Malouf, R. (2002). A comparison of algorithms for maximum entropy parameter estimation. *In Sixth Conf. on Natural Language Learning,* pages 49-55.

McCallum, A., Freitag, D., & Pereira, F. (2000). Maximum entropy Markov models for information extraction and segmentation. *Proc. ICML 2000,* pp. 591-598.

Nie, Z., Zhang, Y., Wen, J. R., & Ma, W. Y. (2005). Object-Level Ranking: Bringing Order to Web Objects. *WWW2005*.

Peng, F., & McCallum, A. (2004). Accurate Information Extraction from Research Papers using Conditional Random Fields. *Proceedings of Human Language Technology Conference and North American Chapter of the Association for Computational Linguistics*.

Sha, F., & Pereira, F. (2003). Shallow Parsing with Conditional Random Fields. *Proceedings of Human Language Technology, NAACL 2003*.

Sutton, C., Rohanimanesh, K., & McCallum, A. (2004). Dynamic Conditional Random Fields: Factorized Probabilistic Models for Labeling and Segmenting Sequence Data. *Proc.ICML2004*.

Taskar, B., Abbeel, P., & Koller, D. (2002). Discriminative probabilistic models for relational data. *In Proceedings of 18th Conference on Uncertainty in Artificial Intelligence (UAI-02),* pp. 485-492, Edmonton, Canada.

Zhai Y., and Liu B. (2005).Web Data Extraction Based on Partial Tree Alignment. *WWW2005.*