Decoding in Latent Conditional Models: A Practically Fast Solution for a NP-hard Problem

Xu Sun[†] **Jun'ichi Tsujii**[†‡§]
[†]Department of Computer Science, University of Tokyo, Japan
[‡]School of Computer Science, University of Manchester, UK
[§]National Centre for Text Mining, Manchester, UK
{sunxu, tsujii}@is.s.u-tokyo.ac.jp

Abstract

Latent conditional models have become popular recently in both natural language processing and vision processing communities. However, establishing an effective and efficient inference method on latent conditional models remains a question. Actually, inference in graphical models, even in a linear chain case (the case discussed in this work), is NP-hard. In this paper, we describe the latent-dynamic inference (LDI), which is able to produce the optimal label sequence on latent conditional models by using efficient search strategy and dynamic programming. Furthermore, we describe a straightforward solution on approximating the LDI, and show that the approximated LDI performs as well as the exact LDI, while the speed is much faster. Our experiments demonstrate that the proposed inference algorithm outperforms existing inference methods on a variety of natural language processing tasks.[1]

1 Introduction

When data have distinct sub-structures, models exploiting latent variables are advantageous in learning (Matsuzaki et al., 2005; Petrov and Klein, 2007; Blunsom et al., 2008). Actually, discriminative probabilistic latent variable models (DPLVMs) have recently become popular choices for performing a variety of tasks with sub-structures, e.g., vision recognition (Morency et al., 2007), syntactic parsing (Petrov and Klein, 2008), and syntactic chunking (Sun et al., 2008). Morency et al. (Morency et al., 2007) demonstrated that DPLVM models could efficiently learn sub-structures of natural problems, and outperform several widely-used conventional models, e.g., support vector machines (SVMs), conditional random fields (CRFs) and hidden Markov models (HMMs). Petrov and Klein (Petrov and Klein, 2008) reported on a syntactic parsing task that DPLVM models can learn more compact and accurate grammars than the conventional techniques without latent variables. The effectiveness of DPLVMs was also shown on a syntactic chunking task by Sun et al. (Sun et al., 2008).

DPLVMs outperform conventional learning models, as described in the aforementioned publications. However, inferences on the latent conditional models are remaining problems. In conventional models such as CRFs, the optimal label path can be efficiently obtained by the dynamic programming. However, for latent conditional models such as DPLVMs, the inference is not straightforward because of the inclusion of latent variables.

In this paper, we propose a new inference algorithm, latent dynamic inference (LDI), by systematically combining an efficient search strategy with the dynamic programming. The LDI is an exact inference method producing the most probable label sequence. In addition, we also propose an approximated LDI algorithm for faster speed. We show that the approximated LDI performs as well as the exact one. We will also discuss a post-processing method for the LDI algorithm: the

[1] Technical Report of the 1st workshop on Latent Dynamics (Jun 16 2010, Tokyo, Japan). Materials of this Technical Report are from a published conference paper in proceedings of European association of computational linguistics 2009 (EACL 2009). For more details of the work, refer to "Sequential Labeling with Latent Variables: An Exact Inference Algorithm and Its Efficient Approximation", Xu Sun and Jun'ichi Tsujii, EACL 2009.

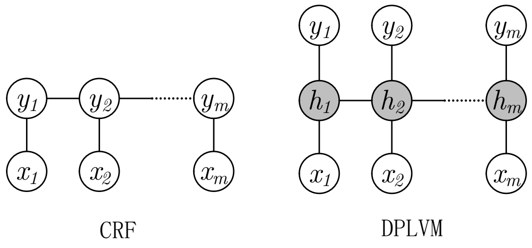

Figure 1: Comparison between CRF models and DPLVM models on the training stage. x represents the observation sequence, y represents labels and h represents the latent variables assigned to the labels. Note that only the white circles are observed variables. Also, only the links with the current observations are shown, but for both models, long range dependencies are possible.

minimum bayesian risk reranking.

The subsequent section describes an overview of DPLVM models. We discuss the probability distribution of DPLVM models, and present the LDI inference in Section 3. Finally, we report experimental results and begin our discussions in Section 4 and Section 5.

2 Discriminative Probabilistic Latent Variable Models

Given the training data, the task is to learn a mapping between a sequence of observations $\mathbf{x} = x_1, x_2, \ldots, x_m$ and a sequence of labels $\mathbf{y} = y_1, y_2, \ldots, y_m$. Each y_j is a class label for the j'th token of a word sequence, and is a member of a set $\mathbf{Y}$ of possible class labels. For each sequence, the model also assumes a sequence of latent variables $\mathbf{h} = h_1, h_2, \ldots, h_m$, which is unobservable in training examples.

The DPLVM model is defined as follows (Morency et al., 2007):

$$P(\mathbf{y}|\mathbf{x}, \mathbf{\Theta}) = \sum_{\mathbf{h}} P(\mathbf{y}|\mathbf{h}, \mathbf{x}, \mathbf{\Theta}) P(\mathbf{h}|\mathbf{x}, \mathbf{\Theta}), \quad (1)$$

where $\mathbf{\Theta}$ represents the parameter vector of the model. DPLVM models can be seen as a natural extension of CRF models, and CRF models can be seen as a special case of DPLVMs that employ only one latent variable for each label.

To make the training and inference efficient, the model is restricted to have disjointed sets of latent variables associated with each class label. Each h_j is a member in a set $\mathbf{H}_{y_j}$ of possible latent variables for the class label y_j. $\mathbf{H}$ is defined as the set of all possible latent variables, i.e., the union of all $\mathbf{H}_{y_j}$ sets. Since sequences which have any $h_j \notin \mathbf{H}_{y_j}$ will by definition have $P(\mathbf{y}|h_j, \mathbf{x}, \mathbf{\Theta}) = 0$, the model can be further defined as:

$$P(\mathbf{y}|\mathbf{x}, \mathbf{\Theta}) = \sum_{\mathbf{h} \in \mathbf{H}_{y_1} \times \ldots \times \mathbf{H}_{y_m}} P(\mathbf{h}|\mathbf{x}, \mathbf{\Theta}), \quad (2)$$

where $P(\mathbf{h}|\mathbf{x}, \mathbf{\Theta})$ is defined by the usual conditional random field formulation:

$$P(\mathbf{h}|\mathbf{x}, \mathbf{\Theta}) = \frac{\exp \mathbf{\Theta} \cdot \mathbf{f}(\mathbf{h}, \mathbf{x})}{\sum_{\forall \mathbf{h}} \exp \mathbf{\Theta} \cdot \mathbf{f}(\mathbf{h}, \mathbf{x})}, \quad (3)$$

in which $\mathbf{f}(\mathbf{h}, \mathbf{x})$ is a feature vector. Given a training set consisting of n labeled sequences, $(\mathbf{x}_i, \mathbf{y}_i)$, for $i = 1 \ldots n$, parameter estimation is performed by optimizing the objective function,

$$L(\mathbf{\Theta}) = \sum_{i=1}^{n} \log P(\mathbf{y}_i|\mathbf{x}_i, \mathbf{\Theta}) - R(\mathbf{\Theta}). \quad (4)$$

The first term of this equation represents a conditional log-likelihood of a training data. The second term is a regularizer that is used for reducing overfitting in parameter estimation.

3 Latent-Dynamic Inference

On latent conditional models, marginalizing latent paths exactly for producing the optimal label path is a computationally expensive problem. Nevertheless, we had an interesting observation on DPLVM models that they normally had a highly concentrated probability mass, i.e., the major probability are distributed on top-n ranked latent paths.

Figure 2 shows the probability distribution of a DPLVM model using a L_2 regularizer with the variance $\sigma^2 = 1.0$. As can be seen, the probability distribution is highly concentrated, e.g., 90% of the probability is distributed on top-800 latent paths.

Based on this observation, we propose an inference algorithm for DPLVMs by efficiently combining search and dynamic programming.

3.1 LDI Inference

In the inference stage, given a test sequence $\mathbf{x}$, we want to find the most probable label sequence, $\mathbf{y}^*$:

$$\mathbf{y}^* = \text{argmax}_{\mathbf{y}} P(\mathbf{y}|\mathbf{x}, \mathbf{\Theta}^*). \quad (5)$$

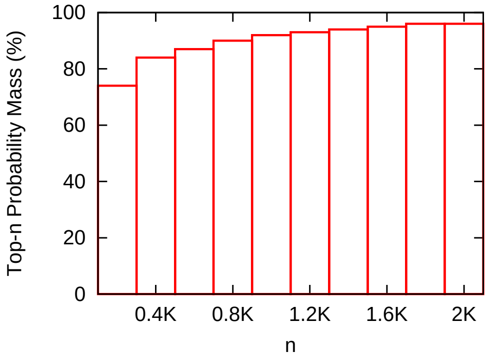

Figure 2: The probability mass distribution of latent conditional models on a NP-chunking task. The horizontal line represents the n of top-n latent paths. The vertical line represents the probability mass of the top-n latent paths.

For latent conditional models like DPLVMs, the $\mathbf{y}^*$ cannot directly be produced by the Viterbi algorithm because of the incorporation of latent variables.

In this section, we describe an exact inference algorithm, the latent-dynamic inference (LDI), for producing the optimal label sequence $\mathbf{y}^*$ on DPLVMs (see Figure 3). In short, the algorithm generates the best latent paths in the order of their probabilities. Then it maps each of these to its associated label paths and uses a method to compute their exact probabilities. It can continue to generate the next best latent path and the associated label path until there is not enough probability mass left to beat the best label path.

In detail, an A^* search algorithm[2] (Hart et al., 1968) with a Viterbi heuristic function is adopted to produce top-n latent paths, $\mathbf{h}_1, \mathbf{h}_2, \dots \mathbf{h}_n$. In addition, a forward-backward-style algorithm is used to compute the exact probabilities of their corresponding label paths, $\mathbf{y}_1, \mathbf{y}_2, \dots \mathbf{y}_n$. The model then tries to determine the optimal label path based on the top-n statistics, without enumerating the remaining low-probability paths, which could be exponentially enormous.

The optimal label path y^* is ready when the following "exact-condition" is achieved:

$$P(\mathbf{y}_1|\mathbf{x},\mathbf{\Theta})-(1-\sum_{\mathbf{y}_k\in\mathbf{LP}_n} P(\mathbf{y}_k|\mathbf{x},\mathbf{\Theta})) \geq 0, \quad (6)$$

```
Definition:
$\mathrm{Proj}(\mathbf{h}) = \mathbf{y} \iff h_j \in \mathbf{H}_{y_j}$ for $j = 1 \dots m$;
$P(\mathbf{h}) = P(\mathbf{h}|\mathbf{x},\mathbf{\Theta})$;
$P(\mathbf{y}) = P(\mathbf{y}|\mathbf{x},\mathbf{\Theta})$.
Input:
weight vector $\mathbf{\Theta}$, and feature vector $F(\mathbf{h},\mathbf{x})$.
Initialization:
$\mathrm{Gap} = -1$; $n = 0$; $P(\mathbf{y}^*) = 0$; $\mathbf{LP}_0 = \emptyset$.
Algorithm:

while $\mathrm{Gap} < 0$ do
    $n = n + 1$
    $\mathbf{h}_n = \mathrm{HeapPop}[\mathbf{\Theta}, F(\mathbf{h},\mathbf{x})]$
    $\mathbf{y}_n = \mathrm{Proj}(\mathbf{h}_n)$
    if $\mathbf{y}_n \notin \mathbf{LP}_{n-1}$ then
        $P(\mathbf{y}_n) = \mathrm{DynamicProg} \sum_{\mathbf{h}:\mathrm{Proj}(\mathbf{h})=\mathbf{y}_n} P(\mathbf{h})$
        $\mathbf{LP}_n = \mathbf{LP}_{n-1} \cup \{\mathbf{y}_n\}$
        if $P(\mathbf{y}_n) > P(\mathbf{y}^*)$ then
            $\mathbf{y}^* = \mathbf{y}_n$
        $\mathrm{Gap} = P(\mathbf{y}^*) - (1 - \sum_{\mathbf{y}_k\in\mathbf{LP}_n} P(\mathbf{y}_k))$
    else
        $\mathbf{LP}_n = \mathbf{LP}_{n-1}$
Output:
the most probable label sequence $\mathbf{y}^*$.
```

Figure 3: The exact LDI inference for latent conditional models. In the algorithm, HeapPop means popping the next hypothesis from the A^* heap; By the definition of the A^* search, this hypothesis (on the top of the heap) should be the latent path with maximum probability in current stage.

where $\mathbf{y}_1$ is the most probable label sequence in current stage. It is straightforward to prove that $\mathbf{y}^* = \mathbf{y}_1$, and further search is unnecessary. This is because the remaining probability mass, $1-\sum_{\mathbf{y}_k\in\mathbf{LP}_n} P(\mathbf{y}_k|\mathbf{x},\mathbf{\Theta})$, cannot beat the current optimal label path in this case.

A simple proof

Given the *exact condition*

$$P(\mathbf{y}_1|\mathbf{x},\mathbf{\Theta})-(1-\sum_{\mathbf{y}_k\in\mathbf{LP}_n} P(\mathbf{y}_k|\mathbf{x},\mathbf{\Theta})) \geq 0, \quad (7)$$

suppose there is a label sequence $\mathbf{y}'$ with a larger probability,

$$P(\mathbf{y}'|\mathbf{x},\mathbf{\Theta}) > P(\mathbf{y}_1|\mathbf{x},\mathbf{\Theta}), \quad (8)$$

then it follows that $\mathbf{y}' \notin \mathbf{LP}_n$, because otherwise it will happen that

$$P(\mathbf{y}'|\mathbf{x},\mathbf{\Theta}) \leq P(\mathbf{y}_1|\mathbf{x},\mathbf{\Theta}) = \max_{\mathbf{y}_k\in\mathbf{LP}_n} P(\mathbf{y}_k|\mathbf{x},\mathbf{\Theta}). \quad (9)$$

[2] A^* search and its variants, like beam-search, are widely used in statistical machine translation. Compared to other search techniques, an interesting point of A^* search is that it can produce top-n results one-by-one in an efficient manner.

It follows that

$$\begin{aligned}
&P(\mathbf{y}'|\mathbf{x},\mathbf{\Theta}) + \sum_{\mathbf{y}_k \in \mathbf{LP}_n} P(\mathbf{y}_k|\mathbf{x},\mathbf{\Theta}) \\
&> P(\mathbf{y}_1|\mathbf{x},\mathbf{\Theta}) + \sum_{\mathbf{y}_k \in \mathbf{LP}_n} P(\mathbf{y}_k|\mathbf{x},\mathbf{\Theta}) \\
&\geq (1 - \sum_{\mathbf{y}_k \in \mathbf{LP}_n} P(\mathbf{y}_k|\mathbf{x},\mathbf{\Theta})) + \sum_{\mathbf{y}_k \in \mathbf{LP}_n} P(\mathbf{y}_k|\mathbf{x},\mathbf{\Theta}) \\
&= 1.
\end{aligned} \tag{10}$$

Therefore, we have

$$P(\mathbf{y}'|\mathbf{x},\mathbf{\Theta}) + \sum_{\mathbf{y}_k \in \mathbf{LP}_n} P(\mathbf{y}_k|\mathbf{x},\mathbf{\Theta}) > 1, \tag{11}$$

which is impossible, therefore the assumption of $\mathbf{y}'$ is impossible.

3.2 An Approximated Version of the LDI

By simply setting a threshold value on the search step, n, we can approximate the LDI, i.e., LDI-Approximation (LDI-A). This is a quite straightforward method for approximating the LDI. In fact, we have also tried other methods for approximation. Intuitively, one alternative method is to design an approximated "exact condition" by using a factor, α, to estimate the distribution of the remaining probability:

$$P(\mathbf{y}_1|\mathbf{x},\mathbf{\Theta}) - \alpha(1 - \sum_{\mathbf{y}_k \in \mathbf{LP}_n} P(\mathbf{y}_k|\mathbf{x},\mathbf{\Theta})) \geq 0. \tag{12}$$

For example, if we believe that at most 50% of the unknown probability, $1 - \sum_{\mathbf{y}_k \in \mathbf{LP}_n} P(\mathbf{y}_k|\mathbf{x},\mathbf{\Theta})$, can be distributed on a single label path, we can set $\alpha = 0.5$ to make a loose condition to stop the inference. At first glance, this seems to be quite natural. However, when we compared this alternative method with the aforementioned approximation on search steps, we found that it worked worse than the latter, in terms of performance and speed. Therefore, we focus on the approximation on search steps in this paper.

References

Phillip Blunsom, Trevor Cohn, and Miles Osborne. 2008. A discriminative latent variable model for statistical machine translation. *Proceedings of ACL'08*.

P.E. Hart, N.J. Nilsson, and B. Raphael. 1968. A formal basis for the heuristic determination of minimum cost path. *IEEE Trans. On System Science and Cybernetics*, SSC-4(2):100–107.

Takuya Matsuzaki, Yusuke Miyao, and Jun'ichi Tsujii. 2005. Probabilistic CFG with latent annotations. *Proceedings of ACL'05*.

Louis-Philippe Morency, Ariadna Quattoni, and Trevor Darrell. 2007. Latent-dynamic discriminative models for continuous gesture recognition. *Proceedings of CVPR'07*, pages 1–8.

Slav Petrov and Dan Klein. 2007. Improved inference for unlexicalized parsing. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL'07)*, pages 404–411, Rochester, New York, April. Association for Computational Linguistics.

Slav Petrov and Dan Klein. 2008. Discriminative log-linear grammars with latent variables. In J.C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20 (NIPS)*, pages 1153–1160, Cambridge, MA. MIT Press.

Xu Sun, Louis-Philippe Morency, Daisuke Okanohara, and Jun'ichi Tsujii. 2008. Modeling latent-dynamic in shallow parsing: A latent conditional model with improved inference. *Proceedings of the 22nd International Conference on Computational Linguistics (COLING'08)*, pages 841–848.