Maximum Reconstruction Estimation for Generative Latent-Variable Models

Yong Cheng#, Yang Liu† * and Wei Xu#

#Institute for Interdisciplinary Information Sciences, Tsinghua University, Beijing, China
†State Key Laboratory of Intelligent Technology and Systems
Tsinghua National Laboratory for Information Science and Technology
Department of Computer Science and Technology, Tsinghua University, Beijing, China
chengyong3001@gmail.com
(liuyang2011, weixu)@tsinghua.edu.cn

Abstract

Generative latent-variable models are important for natural language processing but are limited by the need to learn latent structures from unlabeled data. In this work, we introduce maximum reconstruction estimation (MRE) to learn generative latent-variable models. The basic idea is to circumvent irrelevant but common correlations by maximizing the probability of reconstructing observed data. Our approach keeps the same tractability of learning two classic generative latent-variable models as its MLE counterpart. In this work, we apply MRE to learning two classic generative latent-variable models (i.e., HMMs and IBM translation models) and design efficient dynamic programming algorithms to calculate expectations exactly.

Introduction

The need to learn latent structures from unlabeled data arises in many different problems in natural language processing (NLP), including part-of-speech (POS) induction (Merialdo 1994; Johnson 2007), word alignment (Brown et al. 1993; Vogel, Ney, and Tillmann 1996), syntactic parsing (Klein and Manning 2004; Smith and Eisner 2005), and semantic parsing (Poon and Domingos 2009). Generative latent-variable models such as hidden Markov models (HMMs) and latent-variable probabilistic context-free grammars (LPCFGs) have been widely used for unsupervised structured prediction due to their capability of providing compact representations of data.

Maximum likelihood estimation (MLE) is the standard criterion for training generative latent-variable models: maximizing the likelihood of observed data by marginalizing over latent variables, typically via the Expectation Maximization (EM) algorithm. However, recent studies have revealed that MLE suffers from a significant problem: it may guide the model to focus on explaining irrelevant but common correlations in the data (Ganchev et al. 2010). For example, in unsupervised POS induction, estimating tag probabilities are often prone to be disrupted by high-frequency words such as “,” and “is” (see Table 3).

In this work, we introduce maximum reconstruction estimation (MRE) (Hinton, Osindero, and Teh 2006; Vincent et al. 2008; Bengio 2009; Vincent et al. 2010; Socher et al. 2011; Ammar, Dyer, and Smith 2014; Alain et al. 2015) for learning generative latent-variable models. The new objective is based on an encoding-reconstruction framework: first generating a latent structure conditioned on the observed data (encoding) and then re-generating the observation based on the latent structure (reconstruction). Our approach has the following advantages:

1. Direct learning of model parameters. As we are only interested in learning the parameters of data distribution (i.e., \( P(x; \theta) \)), it is challenging to define the reconstruction distribution (i.e., \( P(x|\theta) \)) using data distribution parameters (i.e., \( \theta \)) due to the intricate dependencies between sub-structures in generative latent-variable models. While previous work has to maintain separate sets of model parameters for encoding and reconstruction to facilitate efficient inference (Ammar, Dyer, and Smith 2014), our approach is capable of directly learning intended model parameters.

2. Tractable inference. Our approach keeps the same tractability of learning generative latent-variable models as its MLE counterpart in this work, we apply MRE to learning two classic generative latent-variable models (i.e., HMMs and IBM translation models) and design efficient dynamic programming algorithms to calculate expectations exactly.

We evaluate our approach on unsupervised POS induction and unsupervised word alignment. Experiments show that MRE enables generative latent-variable models to better discover intended correlations in data and thus leads to significant improvements over MLE.
Maximum Reconstruction Estimation

Let $x$ be an observation, $z$ be a latent structure, and $P(x; \theta)$ be a generative latent-variable model parameterized by $\theta$:

$$P(x; \theta) = \sum_z P(x, z; \theta)\tag{1}$$

Given a set of training examples $D = \{x^{(s)}\}_{s=1}^S$, the standard maximum likelihood estimation (MLE) criterion aims to find a set of parameters that maximizes the log-likelihood of the training data:

$$\theta^*_{\text{MLE}} = \arg \max_{\theta} \left\{ \sum_{s=1}^S \log P(x^{(s)}; \theta) \right\} \tag{2}$$

After learning model parameters, the Viterbi test-time predictions can be calculated as follows:

$$z^*_{\text{MLE}} = \arg \max_z \left\{ P(x, z; \theta) \right\} \tag{3}$$

Although MLE has been widely used in learning latent variable models, Ganchev et al. (2010) indicate that it may guide the model to focus on explaining irrelevant but common (rather than intended) correlations in data.

Alternatively, we are interested in modeling the probability of reconstructing observations via latent structures:

$$P(\hat{x}|x; \theta) = \sum_z P(z|x; \theta) \cdot P(\hat{x}|x, z; \theta)\tag{4}$$

$$= \mathbb{E}_{z|x; \theta} \left[ P(\hat{x}|x, z; \theta) \right] \tag{5}$$

where $P(z|x; \theta)$ is an encoding sub-model that generates a latent structure $z$ given the observation $x$, and $P(\hat{x}|x, z; \theta)$ is a reconstruction sub-model that reconstructs the observation $\hat{x}$ given the just generated latent structure $z$ as well as the observation itself. Often, we ignore the dependency on the observation itself and simply write the reconstruction submodel as $P(\hat{x}|z; \theta)$. Note that unlike Ammar et al. (2014), we require that both the encoding and reconstruction sub-models share the original model parameters to be learned. This forces training algorithms to focus on learning intended models.

As a result, maximum reconstruction estimation (Hinton, Osindero, and Teh 2006; Vincent et al. 2008; Bengio 2009; Vincent et al. 2010; Socher et al. 2011; Ammar, Dyer, and Smith 2014; Alain et al. 2015) aims to find a set of parameters that maximize the product of reconstruction probabilities of the training data:

$$\theta^*_{\text{MRE}} = \arg \max_{\theta} \left\{ \sum_{s=1}^S \log P(\hat{x}^{(s)}|x^{(s)}; \theta) \right\} \tag{6}$$

We use the exponentiated gradient algorithm (Kivinen and Warmuth 1997) with adaptive learning rate (Bagos, Liakopoulos, and Hamodrakas 2004) for estimating model parameters. The decision rule for computing Viterbi test-time predictions is given by

$$z^*_{\text{MRE}} = \arg \max_z \left\{ P(\hat{x}|z; \theta)P(x|z; \theta) \right\} \tag{7}$$

We aim to reconstruct $x$ in the following sections which indicates $\hat{x} = x$. For simplicity, we do not differentiate $\hat{x}$ and $x$ explicitly, and donate $\hat{x}$ as $x$.

**Applications to Generative Latent-Variable Models**

We apply the maximum reconstruction criterion to training two classical generative latent-variable models: hidden Markov models for unsupervised POS induction (Section 3.1) and IBM translation models for unsupervised word alignment (Section 3.2).

**Hidden Markov Models for Unsupervised POS Induction**

Hidden Markov models (HMMs) are a widely used statistical tool for modeling latent sequential structures. Figure 1 shows an example of unsupervised part-of-speech induction. Given a natural language sentence “Obama made a speech yesterday”, HMMs can be used to infer the latent sequence of part-of-speech tags “NNP VBD DT NN NN”.

More formally, let $x = x_1, \ldots, x_n, \ldots, x_N$ be an observed natural language sentence with $N$ words and $z = z_1, \ldots, z_n, \ldots, z_N$ be the corresponding sequence of part-of-speech tags. We use $x$ and $z$ to denote a single word and tag, respectively. The HMM for unsupervised POS induction is given by

$$P(x; \theta) = \sum_z p(z_1)p(x_1|z_1) \prod_{n=2}^N p(z_n|z_{n-1})p(x_n|z_n)\tag{8}$$

where $p(z)$ is initial probability, $p(z'|z)$ is transition probability, and $p(x|z)$ is emission probability. These parameters are required to be normalized: $\sum_z p(z) = 1$, $\forall z : \sum_{z'} p(z'|z) = 1$, and $\forall z : \sum_z p(x|z) = 1$.

**Maximum Likelihood Estimation**

The partial derivatives of $\log P(x; \theta)$ with respect to model parameters are

$$\frac{\partial \log P(x; \theta)}{\partial p(z)} = \frac{1}{p(z)} \mathbb{E}_{z|x; \theta} \left[ \delta(z_1, z) \right] \tag{9}$$

$$\frac{\partial \log P(x; \theta)}{\partial p(z')} = \frac{1}{p(z'|z)} \mathbb{E}_{z'|x; \theta} \left[ \sum_{n=2}^N \delta(z_{n-1}, z) \delta(z_n, z') \right] \tag{10}$$

$$\frac{\partial \log P(x; \theta)}{\partial p(x|z)} = \frac{1}{p(x|z)} \mathbb{E}_{z|x; \theta} \left[ \sum_{n=1}^N \delta(x_n, x) \delta(z_n, z) \right] \tag{11}$$
Clearly, maximum likelihood estimators focus on calculating event expectations (e.g., the expected count of the event that the first tag \( z_1 \) happens to be \( z \)). These expectations can be exactly and efficiently calculated using dynamic programming algorithms.

**Maximum Reconstruction Estimation** The reconstruction probability for HMMs can be written as

\[
P(x|x; \theta) = \sum_z P(z|x; \theta)P(x|z; \theta) \tag{12}
\]

Note that \( P(x|z; \theta) = \prod_{n=1}^{N} p(x_n|z_n) \) is readily defined in HMMs. Since the posterior probability \( P(z|x; \theta) \) is not straightforward to calculate, we resort to the Bayes theorem instead:

\[
P(z; \theta) = \frac{P(z; \theta)P(x|z; \theta)}{P(x; \theta)} \tag{13}
\]

where \( P(z; \theta) = p(z_1) \prod_{n=2}^{N} p(z_n|z_{n-1}) \).

Therefore, Eq. (12) can be re-written as

\[
P(x|x; \theta) = \sum_{z} P(z; \theta)P(x|z; \theta)^2 \tag{14}
\]

Note that all terms in the right-hand side of Eq. (14) are differentiable with respect to model parameters now. This is critical for directly learning data distribution parameters using the MRE criterion.

The partial derivatives of \( \log P(x|x; \theta) \) with respect to model parameters are

\[
\frac{\partial \log P(x|x; \theta)}{\partial p(z)} = \frac{1}{p(z)} \left( E_Q [\delta(z_1, z)] - E_{z|x; \theta} [\delta(z_1, z)] \right) \tag{15}
\]

\[
\frac{\partial \log P(x|x; \theta)}{\partial p(z'|z)} = \frac{1}{p(z'|z)} \left( E_Q \left[ \sum_{n=2}^{N} \delta(z_{n-1}, z)\delta(z_n, z') \right] - E_{z|x; \theta} \left[ \sum_{n=2}^{N} \delta(z_{n-1}, z)\delta(z_n, z') \right] \right) \tag{16}
\]

\[
\frac{\partial \log P(x|x; \theta)}{\partial p(x|z)} = \frac{1}{p(x|z)} \left( E_{z|x; \theta} \left[ \sum_{n=1}^{N} 2\delta(x_n, x)\delta(z_n, z) \right] - E_{z|x; \theta} \left[ \sum_{n=1}^{N} \delta(x_n, z)\delta(z_n, z) \right] \right) \tag{17}
\]

where the distribution \( Q \) is defined as

\[
Q(x, z; \theta) = \frac{P(z; \theta)P(x|z; \theta)^2}{\sum_{z'} P(z'; \theta)P(x|z'; \theta)^2} \tag{18}
\]

![Figure 2: Word alignment. Given a (romanized) Chinese sentence and an English sentence, the task is to identify the latent correspondence between Chinese and English words.](image)

While MLE calculates the expected count of an event with respect to the posterior distribution, MRE calculates the difference between the expected count with respect to the \( Q \) distribution and that with respect to the posterior distribution. Since the difference can be negative, we use the exponentiated gradient algorithm (Kivinen and Warmuth 1997) with adaptive learning rate (Bagos, Liakopoulos, and Hamodrakas 2004) for estimating model parameters. The expectations with respect to \( Q \) can also be exactly and efficiently calculated using dynamic programming algorithms (see Appendix A).

**IBM Translation Models for Unsupervised Word Alignment**

IBM translation models (Brown et al. 1993) provide a principal mechanism for translating between natural languages. The latent structure in IBM models is *word alignment*, which identifies the correspondence between words in two languages. For example, Figure 2 shows a (romanized) Chinese sentence “aobama zuotian fabiao le yanjiang” and an English sentence “Obama made a speech yesterday”. IBM models can be used to compute the latent correspondence between Chinese and English words. Note that we follow Ammar, Dyer, and Smith (2014) to treat the Chinese sentence as observation and the English sentence as side information.

More formally, let \( x = x_1, \ldots, x_n, \ldots, x_N \) be a foreign language sentence with \( N \) words and \( y = y_1, \ldots, y_m, \ldots, y_M \) be an English sentence with \( M \) words. As Brown et al. (1993) restrict that each foreign word is aligned to exactly one English word, an alignment is defined as \( z = z_1, \ldots, z_n, \ldots, z_N \), where \( z_n \in \{ 0, 1, \ldots, M \} \). For example, \( z_n = m \) denotes that \( x_n \) connects to \( y_m \). Note that \( y_0 \) is an empty ciept. For example, in Figure 2, the Chinese word “le” is aligned to an empty ciept (i.e., unaligned graphically).

In this work, we focus on IBM Model 2:

\[
P(x|y; \theta) = \sum_z P(x, z|y; \theta) \tag{19}
\]

\[
= \sum_z \epsilon \prod_{n=1}^{N} p(z_n|n, M, N)p(x_n|y_{z_n}) \tag{20}
\]

where \( p(x|y) \) is *translation probability*, \( p(m'|n', M', N') \) is *alignment probability*, and \( \epsilon \) is a small fixed number.
for the length sub-model. These parameters are required to be normalized: \( \forall y : \sum_x p(x|y) = 1 \) and \( \forall n', M', N' : \sum_{m'} p(m'|n', M', N') = 1 \).

**Maximum Likelihood Estimation** The partial derivatives of log \( P(x|y) \) with respect to model parameters are

\[
\frac{\partial \log P(x|y)}{\partial p(x|y)} = \frac{1}{p(x|y)} \mathbb{E}_z \left[ \sum_{n=1}^{N} \delta(x_n, x) \delta(y_{z_n}, y) \right]
\]

(21)

\[
\frac{\partial \log P(x|y)}{\partial p(m'|n', M', N')} = \frac{\delta(M', M) \delta(N', N)}{p(m'|n', M', N')} \mathbb{E}_{z} \left[ \sum_{n=1}^{N} \delta(z_n, m') \delta(n, n') \right]
\]

(22)

Please refer to (Brown et al. 1993) for details about calculating the expectations.

**Maximum Reconstruction Estimation** Although both \( x \) and \( y \) are observed, we are only interested in reconstructing \( x \) because of the goal of machine translation: translating \( y \) to \( x \). The reconstruction probability is defined as

\[
P(x|y; \theta) = \sum_z P(z|x, y; \theta) P(x|z, y; \theta) \quad \text{(23)}
\]

\[
= \sum_z P(x, z|y; \theta) P(x, y; \theta) \quad \text{(24)}
\]

where the probability of re-generating \( x \) given \( y \) and \( z \) is given by

\[
P(x|z, y; \theta) = \prod_{n=1}^{N} p(x_n|y_{z_n})
\]

(25)

Note that all terms in Eq. (24) are differentiable with respect to model parameters.

The partial derivatives of log \( P(x|y; \theta) \) with respect to model parameters are

\[
\frac{\partial \log P(x|y; \theta)}{\partial p(x|y)} = \frac{1}{p(x|y)} \mathbb{E}_z \left[ \sum_{n=1}^{N} \delta(x_n, x) \delta(y_{z_n}, y) \right] - \mathbb{E}_{z|x,y,\theta} \left[ \sum_{n=1}^{N} \delta(x_n, x) \delta(y_{z_n}, y) \right]
\]

(26)

\[
= \frac{\delta(M', M) \delta(N', N)}{p(m'|n', M', N')} \times \left( \mathbb{E}_z \left[ \sum_{n=1}^{N} \delta(z_n, m') \delta(n, n') \right] - \mathbb{E}_{z|x,y,\theta} \left[ \sum_{n=1}^{N} \delta(z_n, m') \delta(n, n') \right] \right)
\]

(27)

<table>
<thead>
<tr>
<th># state</th>
<th>MLE accuracy</th>
<th>VI accuracy</th>
<th>MRE accuracy</th>
<th>VI accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>0.4054</td>
<td>3.0575</td>
<td>0.3881</td>
<td>2.9322</td>
</tr>
<tr>
<td>20</td>
<td>0.4804</td>
<td>3.1119</td>
<td>0.5203</td>
<td>2.8879</td>
</tr>
<tr>
<td>30</td>
<td>0.5341</td>
<td>3.0835</td>
<td>0.5653</td>
<td>2.8199</td>
</tr>
<tr>
<td>40</td>
<td>0.5817</td>
<td>3.1780</td>
<td>0.6191</td>
<td>2.9255</td>
</tr>
<tr>
<td>50</td>
<td>0.6108</td>
<td>3.2087</td>
<td>0.6739</td>
<td>2.7522</td>
</tr>
</tbody>
</table>

Table 1: Comparison of MLE and MRE on HMMs for unsupervised part-of-speech induction. The evaluation metrics are many-to-one accuracy (accuracy) and variation of information (VI).

Note that Eq. (28) is equivalent to

\[
Q(x, y, z; \theta) = \frac{P(x, z|y; \theta) P(x|z, y; \theta)}{\sum_{z'} P(x, z'|y; \theta) P(x|z', y; \theta)}
\]

(28)

The expectations with respect to Q can also be exactly and efficiently calculated (see Appendix B).

**Experiments**

We evaluated our approach on two unsupervised NLP tasks: part-of-speech induction and word alignment.

**Evaluation on Part-of-Speech Induction**

**Setting** We split the English Penn Treebank into two parts: 46K sentences for training and test and 1K sentences for optimizing hyper-parameters of the exponentiated gradient (EG) algorithm with adaptive learning rate. Each word is manually labeled with a gold-standard part-of-speech tag.

We used two evaluation metrics: many-to-1 accuracy (Johnson 2007) and variation of information (VI) (Beal 2003). The EM algorithm for maximum likelihood estimation runs for 100 iterations and the EG algorithm with adaptive learning rate runs for 50 iterations with initialization of a basic HMM (Ammar, Dyer, and Smith 2014). The number of hidden states in HMMs is set to 50, which is close to the size of the POS tag set.

**Comparison with MLE** Table 1 shows the comparison of MLE and MRE. With the increase of the number of hidden states, the expressiveness of HMMs generally improves accordingly. We find that MRE outperforms MLE for 50-state HMMs in terms of both many-to-one accuracy and VI, suggesting that our approach is capable of guiding the HMMs to use latent structures to find intended correlations in the data. The differences are statistically significant \( (p < 0.01) \). On average, the reconstruction probability of training examples using model parameters learned by MLE \( (i.e., \tilde{\theta}_{MLE}) \) is \( 10^{84} \). In contrast, the average reconstruction probability by MRE \( (i.e., \tilde{\theta}_{MRE}) \) is \( e^{105} \).

Table 2 gives the results on training corpora of various sizes. Generally, the accuracy improves with the increase of
Table 2: Effect of training corpus size.

<table>
<thead>
<tr>
<th># sent.</th>
<th>MLE</th>
<th></th>
<th>MRE</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>accuracy</td>
<td>VI</td>
<td>accuracy</td>
<td>VI</td>
</tr>
<tr>
<td>10,000</td>
<td>0.5087</td>
<td>3.2471</td>
<td>0.5825</td>
<td>2.9018</td>
</tr>
<tr>
<td>20,000</td>
<td>0.5390</td>
<td>3.2873</td>
<td>0.5874</td>
<td>2.9217</td>
</tr>
<tr>
<td>30,000</td>
<td>0.5556</td>
<td>3.0764</td>
<td>0.6000</td>
<td>2.7904</td>
</tr>
<tr>
<td>40,000</td>
<td>0.5800</td>
<td>3.0117</td>
<td>0.6112</td>
<td>2.7403</td>
</tr>
</tbody>
</table>

Table 3: Example emission probabilities for the POS tag “VBD” (verb past tense).

<table>
<thead>
<tr>
<th>.</th>
<th>MLE</th>
<th></th>
<th>MRE</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>said</td>
<td>0.2077</td>
<td>said</td>
<td>0.4632</td>
<td></td>
</tr>
<tr>
<td>is</td>
<td>0.1514</td>
<td>reported</td>
<td>0.0326</td>
<td></td>
</tr>
<tr>
<td>says</td>
<td>0.0312</td>
<td>officials</td>
<td>0.0198</td>
<td></td>
</tr>
<tr>
<td>say</td>
<td>0.0307</td>
<td>announced</td>
<td>0.0195</td>
<td></td>
</tr>
<tr>
<td>:</td>
<td>0.0237</td>
<td>unit</td>
<td>0.0158</td>
<td></td>
</tr>
<tr>
<td>’s</td>
<td>0.0203</td>
<td>noted</td>
<td>0.0119</td>
<td></td>
</tr>
<tr>
<td>think</td>
<td>0.0169</td>
<td>gained</td>
<td>0.0106</td>
<td></td>
</tr>
<tr>
<td>added</td>
<td>0.0129</td>
<td>told</td>
<td>0.0102</td>
<td></td>
</tr>
<tr>
<td>was</td>
<td>0.0129</td>
<td>court</td>
<td>0.0101</td>
<td></td>
</tr>
</tbody>
</table>

Table 4: Comparison between CRF Autoencoders and MRE on unsupervised part-of-speech induction.

<table>
<thead>
<tr>
<th># state</th>
<th>CRF Autoencoders</th>
<th></th>
<th>MRE</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>accuracy</td>
<td>VI</td>
<td>accuracy</td>
<td>VI</td>
</tr>
<tr>
<td>10</td>
<td>0.4059</td>
<td>2.7145</td>
<td>0.3881</td>
<td>2.9322</td>
</tr>
<tr>
<td>20</td>
<td>0.4657</td>
<td>2.7462</td>
<td>0.5203</td>
<td>2.8879</td>
</tr>
<tr>
<td>30</td>
<td>0.5479</td>
<td>2.9585</td>
<td>0.5653</td>
<td>2.8199</td>
</tr>
<tr>
<td>40</td>
<td>0.5377</td>
<td>3.1048</td>
<td>0.6191</td>
<td>2.9255</td>
</tr>
<tr>
<td>50</td>
<td>0.5662</td>
<td>2.8450</td>
<td>0.6739</td>
<td>2.7522</td>
</tr>
</tbody>
</table>

Table 5: Comparison between MLE and MRE on IBM translation models for unsupervised word alignment. The evaluation metric is alignment error rate (AER).

<table>
<thead>
<tr>
<th>criterion</th>
<th>model</th>
<th>C → E</th>
<th>E → C</th>
</tr>
</thead>
<tbody>
<tr>
<td>MLE</td>
<td>Model 1</td>
<td>43.07</td>
<td>45.89</td>
</tr>
<tr>
<td></td>
<td>Model 2</td>
<td>40.28</td>
<td>42.38</td>
</tr>
<tr>
<td>MRE</td>
<td>Model 1</td>
<td>41.90</td>
<td>45.39</td>
</tr>
<tr>
<td></td>
<td>Model 2</td>
<td>38.33</td>
<td>41.73</td>
</tr>
</tbody>
</table>

Table 6: Example translation probabilities of the Chinese word “wenzhang”.

<table>
<thead>
<tr>
<th>article</th>
<th>MLE</th>
<th></th>
<th>MRE</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>the</td>
<td>0.4932</td>
<td>article</td>
<td>0.5428</td>
<td></td>
</tr>
<tr>
<td>says</td>
<td>0.0586</td>
<td>says</td>
<td>0.0624</td>
<td></td>
</tr>
<tr>
<td>points</td>
<td>0.0293</td>
<td>published</td>
<td>0.0497</td>
<td></td>
</tr>
<tr>
<td>an</td>
<td>0.0263</td>
<td>points</td>
<td>0.0349</td>
<td></td>
</tr>
</tbody>
</table>

Evaluation on Word Alignment

Setting We used the FBIS corpus as the training corpus, which contains 240K Chinese-English parallel sentences with 6.9M Chinese words and 8.9M English words. We used the TsinghuaAligner development and test sets (Liu and Sun 2015), which both contain 450 sentence pairs with gold-standard annotations. The evaluation metric is alignment error rate (AER) (Och and Ney 2003). Both MLE and MRE use the following training scheme: 5 iterations for IBM Model 1 and 5 iterations for IBM Model 2. As IBM Model 1 is a simplified version of IBM Model 2, the parameters of Model 1 at iteration 5 are used to initialize Model 2. We distinguish between two translation directions: Chinese-to-English (C → E) and English-to-Chinese (E → C).

Comparison with MLE Table 5 shows the comparison between MLE and MRE. We find that MRE outperforms MLE for both translation directions. All the differences are statistically significant ($p < 0.01$).

Table 6 shows example translation probabilities (i.e., $p(x|y)$) of the Chinese word “wenzhang” (i.e., “article”). We find that MLE tends to identify frequent words such as “the” and “an” as candidate translations while MRE finds more relevant candidate translations. This finding further confirms that MRE is more robust to common but irrelevant correlations.

Comparison with CRF Autoencoders On the same dataset, CRF Autoencoders achieve much lower AERs: 32.54 for C → E and 29.81 for E → C, respectively. The reason is that CRF Autoencoders are a discriminative latent-variable model capable of including more expressive IBM Model 4 as features. In contrast, our approach focuses on providing better training criterion for generative latent-
variable models such as IBM Model 2. The training time for CRF Autoencoders is about 15 days while it only takes our approach 4 hours. Our generative models can also serve as central features in CRF autoencoders.

**Conclusion**

We have presented maximum reconstruction estimation for training generative latent-variable models such as hidden Markov models and IBM translation models. In the future, we plan to apply our approach to more generative latent-variable models such as probabilistic context-free grammars and explore the possibility of developing new training algorithms that minimize reconstruction errors.

**Calculating Expectations for MRE Training of Hidden Markov Models**

The expectations with respect to \( Q \) can also be exactly and efficiently calculated using dynamic programming algorithms. For example, consider the following expectation:

\[
E_Q \left[ \sum_{n=2}^{N} \delta(z_{n-1}, z) \delta(z_n, z') \right]
\]

\[= \sum_{n=2}^{N} P(z; \theta)P(x|z; \theta)^2 \sum_{n=2}^{N} \delta(z_{n-1}, z) \delta(z_n, z') \]  
(30)

Where we re-define the forward probability \( \alpha_n(z) \) as

\[
\alpha_n(z) = \begin{cases} 
p(z)p(x_1|z_2)^2 & \text{if } n = 1 \\
\sum_{z'} \alpha_{n-1}(z')p(z|z')p(x_n|z)^2 & \text{otherwise}
\end{cases}
\]

and the backward probability \( \beta_n(z) \) as

\[
\beta_n(z) = \begin{cases} 
1 & \text{if } n = 1 \\
\sum_{z'} p(z|z')p(x_{n+1}|z')^2 \beta_{n+1}(z') & \text{otherwise}
\end{cases}
\]

(32)

Therefore, the expectation in the denominator of Eq. (31) can be re-written as

\[
E_{\theta, \theta} \left[ P(x|z; \theta)^2 \right]
\]

\[= \sum_{z} p(z_1) \prod_{n=2}^{N} p(z_n|z_{n-1}) \left( \prod_{n=1}^{N} p(x_n|z_n) \right)^2 \]  
(34)

\[= \sum_{z} \alpha_N(z) \]  
(35)

and the expectation in the numerator of Eq. (31) can be re-written as

\[
E_{\theta, \theta} \left[ \sum_{n=2}^{N} \delta(z_{n-1}, z) \delta(z_n, z') \right]
\]

\[= \sum_{a} \sum_{n=2}^{N} p(z_1) \prod_{n=2}^{N} p(z_n|z_{n-1}) \left( \prod_{n=1}^{N} p(x_n|y_n) \right)^2 \times \sum_{n=2}^{N} \delta(z_{n-1}, z) \delta(z_n, z') \]  
(36)

\[= \sum_{n=2}^{N} \sum_{a} \sum_{n=2}^{N} p(z_1) \prod_{n=2}^{N} p(z_n|z_{n-1}) \left( \prod_{n=1}^{N} p(x_n|z_n) \right)^2 \times \delta(z_{n-1}, z) \delta(z_n, z') \]  
(37)

\[= \sum_{n=2}^{N} \sum_{a} \alpha_{n-1}(z)p(z_1)p(x_n^a|z_n^a) \beta_n(z') \]  
(38)

Similarly, other expectations with respect to the \( Q \) distribution can also be calculated using forward and backward probabilities.

**Calculating Expectations for MRE Training of IBM Model 2**

The expectations with respect to \( Q \) can also be exactly and efficiently calculated. Consider the following expectation:

\[
E_Q \left[ \sum_{n=1}^{N} \delta(x_n, x) \delta(y_{a_n}, y) \right]
\]

\[= \sum_{n=1}^{N} P(x, z; \theta)P(x, z; \theta)^2 \sum_{n=1}^{N} \delta(x_n, x) \delta(y_{a_n}, y) \]  
(39)

The denominator of Eq. (39) can be calculated as

\[
\sum_{a} \sum_{n=1}^{N} P(x, z; \theta)P(x, z; \theta) \sum_{n=1}^{N} \delta(x_n, x) \delta(y_{a_n}, y)
\]

\[= \sum_{a} \sum_{n=1}^{N} p(z_1)p(x_1^a|z_1)p(x_n^a|y_{a_n}) \]  
(40)

\[= \sum_{n=1}^{N} \sum_{a} \sum_{n=1}^{N} p(z_1)p(x_1^a|z_1)p(x_n^a|y_{a_n}) \]  
(41)

The numerator of Eq. (39) can be efficiently calculated as

\[
\sum_{a} \sum_{n=1}^{N} P(x, z; \theta)P(x, z; \theta) \sum_{n=1}^{N} \delta(x_n, x) \delta(y_{a_n}, y)
\]

\[= \sum_{a} \sum_{n=1}^{N} p(z_1)p(x_1^a|z_1)p(x_n^a|y_{a_n}) \]  
(42)

\[= \sum_{n=1}^{N} \sum_{a} \sum_{n=1}^{N} p(z_1)p(x_1^a|z_1)p(x_n^a|y_{a_n}) \]  
(43)

Other expectations with respect to the \( Q \) distribution can also be calculated similarly.

\(^1\)Note that it is possible to apply our approach to IBM Model 4. We leave this for future work.
Acknowledgements

This research is supported by the 863 Program (2015AA015407), the National Natural Science Foundation of China (No. 61522204, 6136136003, 61532001), 1000 Talent Plan grant, Tsinghua Initiative Research Program grants 20151080475, a Google Faculty Research Award and Fund from Online Education Research Center, Ministry of Education (No. 2016ZD102). We sincerely thank the viewers for their valuable suggestions.

References


