2009 10th International Conference on Document Analysis and Recognition

Bayesian Similarity Model Estimation for Approximate Recognized Text Search

Atsuhiro Takasu
National Institute of Informatics
takasu@nii.ac.jp

Abstract

Approximate text search is a basic technique to handle recognized text that contains recognition errors. This paper proposes an approximate string search for recognized text using a statistical similarity model focusing on parameter estimation. The main contribution of this paper is to propose a parameter estimation algorithm using variational Bayesian expectation maximization technique. We applied the obtained model to approximate substring detection problem and experimentally showed that the Bayesian estimation is effective.

1 Introduction

Recently many paper books and journals are scanned to make large digital libraries for preserving and utilizing them. Spoken documents are another important information source of digital archives. As a result, future digital libraries may contain various kinds of recognized data such as OCR text, spoken documents and so on. Since recognized documents inevitably contain recognition errors, we need tools to handle erroneous text as well as improvement of recognizers' accuracy.

When utilizing text, we usually use words included in the document as features. Taghva *et al.* [6] reported how OCR errors affect text retrieval performance. Approximate text search is a basic technology to develop a text utilization system that is robust to recognition errors. It has a long research history, and it is applied to wide range of applications such as record matching [2], spoken document retrieval [5], DNA processing, etc. When using approximate search, we need to define a similarity or distance between strings. The edit distance is a basic metric to measure string similarity. It represents the distance of strings as the minimum number of editing operations, i.e., insertion, deletion and substitution, to convert one string to another. Confusion matrix assigns editing cost based on characters, e.g., it may assign different costs for substituting different characters. Srinvasan et al. used the confusion matrix in spoken document retrieval [5].

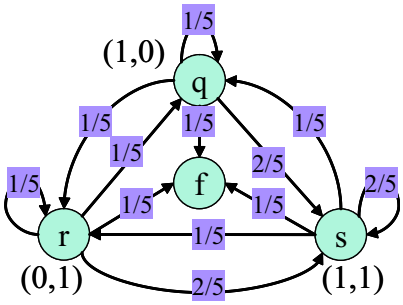
Since digital library system may contain various text obtained from various information sources through various recognizers, we need a method to tune the cost of edit operations according to the text. Therefore, ability to learn recognition error model is effective in digital library. Ristad et al. [4] proposed a learning edit distance that has the ability to learn editing costs from the training data based on the maximum likelihood estimation. We [8] proposed a similar model based on the joint probability of an original string and its recognized one. MacCalum et al. [3] proposed another learning algorithm based on conditional Markov random field model. When learning parameters of a model describing the similarity at the character level we need to estimate $O(n^2)$ parameters where n is the size of alphabet. Maximum Likelihood (ML) estimators have been used for obtaining the models so far [4, 8]. However ML estimators tends to overfit for estimating the models having many parameters, and we need to smooth the estimation.

We have been constructing a digital library that contains millions of scholarly paper documents written in Japanese and/or English. The purpose of this project is to capture academic articles to make an archive of Japanese scholarly information and to utilize them. Currently about 3 million papers are scanned and processed by an OCR. Since Japanese language consists of several thousands of characters, number of parameters required for the model is much more than English, and consequently, smoothing in parameter estimation is crucial problem. Therefore, this paper focuses on the learning problem of string similarity model for approximate text search and proposes a Bayesian learning algorithm to obtain similarity model using the variational Bayesian (VB) technique.

2 Statistical Similarity Model

For statistical string similarity, HMM is often used as a base of the model [4]. Among them, we use the Dual and Variable Length Hidden Markov Model (DVHMM) [8] because it is suitable for handling both Japanese and English characters.

978-0-7695-3725-2/09 $25.00 © 2009 IEEE
DOI 10.1109/ICDAR.2009.193



(a) State transition graph

State q		State r		State s	
Pair	prob.	Pair	prob.	Pair	prob.
(a, λ)	1/2	(λ, a)	1/2	(a, a)	4/9
(b, λ)	1/2	(λ, b)	1/2	(a, b)	1/9
				(b, a)	1/9
				(b, b)	1/3

(b) Output probabilities

Figure 1. Example of DVHMM

This section overviews the DVHMM. The DVHMM is a kind of pairwise hidden Markov model (HMM). It produces a pair of strings by walking around the finite states like HMM and produces a portion of the pair of strings at each state. A state of the DVHMM is characterized by a pair of lengths of the original and observed output strings and the state characterized by (i, j) produces a pair of original and observed strings with lengths i and j, respectively.

Fig. 1 [7] shows an example of a DVHMM that consists of four states. A state q is characterized by $(1, 0)$. From the viewpoint of an editing operation, this state corresponds to a delete operation. A state r is characterized by $(0, 1)$ and it corresponds to an insert operation. A state s is characterized by $(1, 1)$ and it corresponds to a substitution or no edit operation. A state f is the final state where every state transition ends.

Each state except for the final state produces a pair of strings according to the output probability distribution. Suppose the alphabet is $\{a, b\}$. Then, the state q in Fig. 1 produces a pair of strings (a, λ) or (b, λ) where λ stands for a null string. Output probability is assigned to each pair of strings, and the DVHMM produces a pair according to the probabilities. The tables in Fig. 1 (b) show the output probabilities at each state. We denote the output probability distribution of a state s as $\boldsymbol{o}_s$.

The DVHMM has the initial probability distribution which is omitted in Fig. 1 and the transition probabilities, which are denoted as arcs in the figure. We use $\boldsymbol{\pi}$ and $\boldsymbol{\tau}_s$ for a state s to denote the initial and transition probability distributions, respectively. In this way, the costs of the editing operations are represented in the DVHMM as the combination of transition and output probabilities.

The DVHMM defines the joint probability of a pair of strings. Let us consider a pair (ab, aab) of strings and the state transitions $sqsf$ in the DVHMM of Fig. 1. Since s is a substitution and q is an insertion, the pair (ab, aab) is produced by emitting a portion of the pair of strings at each state in the following way:

state	s	q	s
original	a	λ	b
observed	a	a	b

where the second and third rows stand for the original and the observed substrings produced by the state in the first line, respectively. Note that a state transition sequence defines the decomposition of an original and an observed strings. We refer to this decomposition as an *alignment*. Note that for a pair of strings, a state transition sequence uniquely determines an alignment.

For a pair $\boldsymbol{w} \equiv (\boldsymbol{u}, \boldsymbol{v})$ of strings and a state sequence $\boldsymbol{t} = t_1 t_2 \cdots t_l$ generating the pair, the DVHMM defines the joint probability of the pair by the state sequence as

$$\Pr(\boldsymbol{w}, \boldsymbol{t} \mid \boldsymbol{\Theta}) = \pi_{t_1} \prod_{i=1}^{l-1} \tau_{t_i t_{i+1}} \prod_{i=1}^{l} o_{t_i \boldsymbol{w}_i} \qquad (1)$$

where $\boldsymbol{w}_i$ denotes the pair of strings emitted at the ith state t_i, whereas $\boldsymbol{\Theta} \equiv (\boldsymbol{\pi}, \{\boldsymbol{\tau}_s\}_s, \{\boldsymbol{o}_s\}_s)$ is parameters consisting of the initial, transition, and output probabilities. We refer to this joint probability as an *alignment probability*. For the alignment described in the table, the probability is

$$\Pr((ab, aab), sqs) = \pi_s \tau_{sq} \tau_{qs} o_{s(a,a)} o_{q(\lambda,a)} o_{s(b,b)}$$

Generally there are multiple state transitions that produce a given pair of strings. We denote the set of state sequences that produces a pair $\boldsymbol{w}$ as $\mathcal{T}(\boldsymbol{w})$. Then the string similarity of the pair $\boldsymbol{w}$ is defined as the maximum alignment probability

$$\max_{\boldsymbol{t} \in \mathcal{T}(\boldsymbol{w})} \Pr(\boldsymbol{w}, \boldsymbol{t} \mid \boldsymbol{\Theta}) \, . \qquad (2)$$

As described in Section 2.2, each state of a DVHMM corresponds to an edit operation, and a state transition sequence corresponds to edit operations. From this perspective, the state transition that satisfies eq. (2) corresponds to the edit operation sequence with the lowest cost. In this way, the probability defined by Eq. (2) gives similarity to a pair $\boldsymbol{u}$ and $\boldsymbol{v}$ of strings.

By adapting the initial, transition, and output probabilities as well as graphical structure of the model to recognizers, the model can represent the string similarity according to error patterns of various recognizers.

3 Parameter Estimation

This section derives the parameter estimation algorithm based on the Variational Bayesian (VB) technique [1]. Beal proposed VB algorithm for HMM [1]. We extend the algorithm for Bayesian parameter estimation for DVHMM.

Let $\boldsymbol{W} = \{\boldsymbol{w}_1, \boldsymbol{w}_2, \cdots, \boldsymbol{w}_n\}$ be a set of training pairs of strings where $\boldsymbol{w}_i$ $(1 \leq i \leq n)$ is a pair $(\boldsymbol{u}_i, \boldsymbol{v}_i)$ of an original and corresponding recognized strings. For a graphical structure of DVHMM, let us consider the following generative model:

- generate an initial probability distribution $Multi(\boldsymbol{\pi})$ according to a Dirichlet prior $\mathcal{D}(\boldsymbol{\pi}; \boldsymbol{\alpha})$,
- for each state s, generate transition probability distributions $Multi(\boldsymbol{\tau}_s)$ according to a Dirichlet prior $\mathcal{D}(\boldsymbol{\tau}_s; \boldsymbol{\beta}_s)$,
- for each state s, generate output probability distributions $Multi(\boldsymbol{o}_s)$ according to a Dirichlet prior $\mathcal{D}(\boldsymbol{o}_s; \boldsymbol{\gamma}_s)$,
- generate the training pairs $\boldsymbol{W}$ by the DVHMM obtained in the previous steps

where $Multi(\boldsymbol{\theta})$ and $\mathcal{D}(\boldsymbol{\theta}; \boldsymbol{\alpha})$ denote the multinomial and Dirichlet distributions with the parameters $\boldsymbol{\theta}$ and $\boldsymbol{\alpha}$, respectively. Unlike the ordinary HMM, the number of output symbols are different depending on the state. So we need to use different Dirichlet priors for each state. We denote the parameters of Dirichlet priors as

$$\Lambda \equiv (\boldsymbol{\alpha}, \{\boldsymbol{\beta}_s\}_s, \{\boldsymbol{\gamma}_s\}_s) \ .$$

In Bayes parameter estimation, we obtain the probability distributions of parameters of initial, transition, and output probabilities that maximize the following marginal log likelihood

$$\begin{aligned}
\ln \Pr(\boldsymbol{W}; \Lambda) &= \sum_{i=1}^{n} \ln \Pr(\boldsymbol{w}_i; \Lambda) \\
&= \sum_{i=1}^{n} \ln \int \sum_{\boldsymbol{t}_i \in \mathcal{T}(\boldsymbol{w}_i)} q(\boldsymbol{\Theta}, \boldsymbol{T}) \frac{\Pr(\boldsymbol{w}_i, \boldsymbol{t}_i \mid \boldsymbol{\Theta}) \Pr(\boldsymbol{\Theta}; \Lambda)}{q(\boldsymbol{\Theta}, \boldsymbol{T})} d\boldsymbol{\Theta} \\
&\geq \sum_{i=1}^{n} \int \sum_{\boldsymbol{t}_i \in \mathcal{T}(\boldsymbol{w}_i)} q(\boldsymbol{\Theta}, \boldsymbol{T}) \ln \frac{\Pr(\boldsymbol{w}_i, \boldsymbol{t}_i \mid \boldsymbol{\Theta}) \Pr(\boldsymbol{\Theta}; \Lambda)}{q(\boldsymbol{\Theta}, \boldsymbol{T})} d\boldsymbol{\Theta} \\
&\equiv \mathcal{F}(q(\boldsymbol{\Theta}, \boldsymbol{T})) \qquad (3)
\end{aligned}$$

where $\boldsymbol{T}$ is $(\boldsymbol{t}_1, \cdots, \boldsymbol{t}_n)$ and $\boldsymbol{t}_i$ is a random variable for the state sequence for ith training pair. The function $q(\boldsymbol{\Theta}, \boldsymbol{T})$ is any probability density function, i.e.,

$$\int \sum_{\boldsymbol{t}_1 \in \mathcal{T}(\boldsymbol{w}_1)} \cdots \sum_{\boldsymbol{t}_n \in \mathcal{T}(\boldsymbol{w}_n)} q(\boldsymbol{\Theta}, \boldsymbol{T}) d\boldsymbol{\Theta} = 1 \ . \qquad (4)$$

The inequality in Eq. (3) is obtained by Jensen's inequality. To make the learning problem tractable, we factorize the function as follows:

$$q(\boldsymbol{\Theta}, \boldsymbol{T}) \equiv \underbrace{q(\boldsymbol{\pi}) \prod_{s \in S} q(\boldsymbol{\tau}_s) \prod_{s \in S} q(\boldsymbol{o}_s)}_{q(\boldsymbol{\Theta})} \underbrace{\prod_{i=1}^{n} q_i(\boldsymbol{t}_i)}_{q(\boldsymbol{T})} \ . \qquad (5)$$

Variational Bayesian learning algorithm consists of VBE and VBM steps maximizing the lower bound $\mathcal{F}$ with respect to $q_i(\boldsymbol{t}_i)$ and $q(\boldsymbol{\Theta})$, respectively.

In the VBM step, by equating the functional derivatives w.r.t. $q(\pi)$, $q(\tau_s)$, and $q(o_s)$ to 0, we obtain the update formulas.

$$q(\boldsymbol{\pi}) = \mathcal{D}(\boldsymbol{\pi}; \boldsymbol{\alpha}'), \ q(\boldsymbol{\tau}_s) = \mathcal{D}(\boldsymbol{\tau}_s; \boldsymbol{\beta}'_s), \ q(\boldsymbol{o}_s) = \mathcal{D}(\boldsymbol{o}_s; \boldsymbol{\gamma}'_s)$$

where

$$\begin{aligned}
\alpha'_s &\equiv \alpha_s + \frac{1}{n} \sum_{i=1}^{n} \sum_{\boldsymbol{t}_i \in \mathcal{T}(\boldsymbol{w}_i)} q_i(\boldsymbol{t}_i) \delta(s, s_{i1}) \\
\beta'_{sr} &\equiv \beta_{sr} + \frac{1}{n} \sum_{i=1}^{n} \sum_{\boldsymbol{t}_i \in \mathcal{T}(\boldsymbol{w}_i)} q_i(\boldsymbol{t}_i) \mathcal{C}_t(\boldsymbol{t}_i, s, r) \\
\gamma'_{s\boldsymbol{w}} &\equiv \gamma_{s\boldsymbol{w}} + \frac{1}{n} \sum_{i=1}^{n} \sum_{\boldsymbol{t}_i \in \mathcal{T}(\boldsymbol{w}_i)} q_i(\boldsymbol{t}_i) \mathcal{C}_o(\boldsymbol{t}_i, s, \boldsymbol{w}) \qquad (6)
\end{aligned}$$

where $\mathcal{C}_t(\boldsymbol{t}_i, s, r)$ denotes the number of state transition from a state s to r in the state transition sequence $\boldsymbol{t}_i$, whereas $\mathcal{C}_o(\boldsymbol{t}_i, s, \boldsymbol{w})$ denotes the number of emissions of $\boldsymbol{w}$ at a state s in the state transition sequence $\boldsymbol{t}_i$.

In the VBE step, by equating the functional derivertive w.r.t. $q_i(\boldsymbol{t})$ to 0, we obtain

$$q(\boldsymbol{t}_i) \propto \tilde{\pi}_{s_{i1}} \prod_{j=1}^{|\boldsymbol{t}_i|-1} \tilde{\tau}_{t_{ij} t_{ij+1}} \prod_{j=1}^{|\boldsymbol{t}|} \tilde{o}_{t_{ij} \boldsymbol{w}_{ij}} \qquad (7)$$

where

$$\begin{aligned}
\ln \tilde{\pi}_s &\equiv \Psi(\alpha'_s) - \Psi(\sum_{r \in S} \alpha'_r) \\
\ln \tilde{\tau}_{sr} &\equiv \Psi(\beta'_{sr}) - \Psi(\sum_{t \in S} \beta'_{st}) \\
\ln \tilde{o}_{s\boldsymbol{w}} &\equiv \Psi(\gamma'_{s\boldsymbol{w}}) - \Psi(\sum_{\boldsymbol{u} \in \boldsymbol{W}_s} \gamma'_{s\boldsymbol{u}}) \qquad (8)
\end{aligned}$$

where $\Psi(\cdot)$ denotes the digamma function. See Appendix for the sketch of the derivations.

4 Experimental Results

We evaluated the effectiveness of the proposed algorithm through experiments. We used two kinds of data set. One

is abstracts included in papers in our digital library. We selected 1,000 Japanese articles published by Japanese computer science societies and extracted abstracts from them. The other is references included in the same papers. We used 1,000 references for experiments.

For evaluation, we prepared clean text manually for both kinds of data sets. The average recognition accuracy of abstract was 99.2% whereas the accuracy of references was 94.2%. Abstract usually consists of ordinary Japanese text and it contains only a few mathematical expressions. On the other hand, references contain both Japanese and English text. They also contain many punctuations. These features make it hard for OCR to recognize characters. In this experiment, abstracts were selected as ordinary text whereas references were selected as poorly recognized text.

We compared the proposed model with the unit cost edit distance and the same model whose parameters are estimated based on the maximum likelihood estimation. Labels "ED", "ML", and "VBEM" are used for representing the unit cost edit distance, the model obtained by the maximum likelihood estimator, and the model estimated by the proposed variational Bayesian expectation maximization algorithm, respectively.

For each data set, we used half of them as training data and estimated the parameters of the models from them. Remaining half data was used for evaluation. We made 100 queries by extracting words and phrases from the clean test data set. Then, we detected positions of query words and phrases from recognized test data set and ranked them according to the similarities. In approximate search, accuracy is affected by query length. Therefore, we prepared 50 short queries whose average length is 5.3 characters and 50 long queries whose average length is 10.4 characters. Labels "short" and "long" are used for these types of queries, respectively.

As for the evaluation metric, we use the f-measure of the ranked position lists. Figure 2 shows the average f-measures for searching abstracts and references with short and long queries. As shown in the graph, the search accuracy of the three similarities is almost same for the abstracts. Since the OCR recognition accuracy is very high for abstract, the similarity measure does not affect the search performance. On the other hand, the proposed similarity outperforms other similarities for the references for both short and long queries. This is because the learnable string similarity can reflect the OCR error patterns, and proposed smoothed model is effective to obtain similarity model.

5 Conclusion

This paper proposes an approximate string search for recognized text using a statistical similarity model focusing on parameter estimation method. The main contribution of this paper is to propose a Bayesian parameter estimation algorithm using variational Bayesian expectation maximization technique. We applied the obtained model to approximate text search problem in a digital library and showed that the Bayesian estimation is effective for poorly recognized text search.

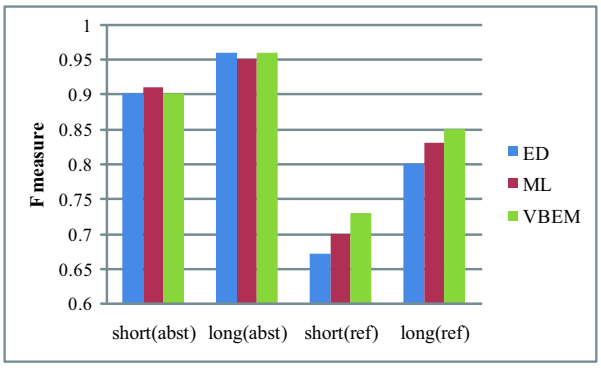


Figure 2. Search performance.

References

[1] M. J. Beal. *"Variational Algorithms for Approximate Baysesian Inference"*. PhD thesis, University College London, 2003.

[2] M. Bilenko and R. J. Mooney. "Adaptive Duplicate Detection Using Learnable String Similarity Measures". In *Proc. of SIGKDD2003*, pp. 39–48, 2003.

[3] A. McCallum and F. Pereira. "A Conditional Random Field fro Discriminatively-trained Finite-state String Edit Similarity". In *Proc. of UAI05*, 2005.

[4] E. S. Ristad and P. N. Yianilos. Learning string-edit distance. *IEEE PAMI*, 20(5):522–532, 1998.

[5] S. Srinivasan and G. Petkovic. Phonetic confusion matrix based spoken document retrieval. In *Proc. of SIGIR00* pp. 81–87, 2000.

[6] K. Taghva, J. Borsack, and A. Condit. Evaluation of model-based retrieval effectiveness with ocr text. *ACM Transaction on Information Systems*, 14(1):64–93, 1996.

[7] A. Takasu. "An Approximate Multi-word Matching Algorithm for Robust Document Retrieval". In *Proc. of CIKM 06*, pages 34–42, 2006.

[8] A. Takasu and K. Aihara. "DVHMM: Variable Length Text Recognition Error Model". In *Proc. of ICPR2002*, pp. 110–114, 2002.

A VBEM Algorithm

To solve the optimization problem, let us introduce Lagrange multipliers λ_π, $\{\lambda_{o_s}\}$, and $\{\lambda_{\tau_s}\}$ for initial, output and transition probabilities, respectively, and λ_i $(1 \leq i \leq n)$ for state sequences. Using Eqs. (3), (5), and these Lagrange multiplies, the Lagrangian is defined as

$$\begin{aligned}
&\mathcal{L}_{VB}(q(\boldsymbol{\Theta}, \boldsymbol{T})) \\
&\equiv \mathcal{F}(q(\boldsymbol{\Theta}, \boldsymbol{T})) + \lambda_\pi \Big(\int q(\boldsymbol{\pi}) d\boldsymbol{\pi} - 1 \Big) + \\
&\quad \sum_{s \in S} \lambda_{\tau_s} \Big(\int q(\boldsymbol{\tau}_s) d\boldsymbol{\tau}_s - 1 \Big) + \sum_{s \in S} \lambda_{o_s} \Big(\int q(\boldsymbol{o}_s) d\boldsymbol{o}_s - 1 \Big) + \\
&\quad \sum_{i=1}^{n} \lambda_i \Big(\sum_{\boldsymbol{t} \in \mathcal{T}(\boldsymbol{w}_i)} q(\boldsymbol{t}_i) - 1 \Big) \qquad (9)
\end{aligned}$$

Variational Bayesian learning algorithm consists of VBE and VBM steps maximizing the lower bound $\mathcal{F}$ with respect to $q_i(\boldsymbol{t}_i)$ and $q(\boldsymbol{\Theta})$, respectively.

A.1 VBM step

In the VBM step, by equating the functional derivatives w.r.t. $q(\boldsymbol{\pi})$, $q(\boldsymbol{\tau}_s)$, and $q(\boldsymbol{o}_s)$ to 0, we obtain the update formulas. For the free distribution for initial probability distribution,

$$\begin{aligned}
&\frac{\partial \mathcal{L}_{VB}}{\partial q(\boldsymbol{\pi})} \\
&= \sum_{i=1}^{n} \sum_{\boldsymbol{t}_i \in \mathcal{T}(\boldsymbol{w}_i)} q(\boldsymbol{t}_i) \ln \Pr(t_{i1} \mid \boldsymbol{\pi}) + \\
&\quad \sum_{i=1}^{n} \{ \ln \Pr(\boldsymbol{\pi}; \boldsymbol{\alpha}) - \ln q(\boldsymbol{\pi}) + 1 \} + \lambda_\pi = 0 \qquad (10)
\end{aligned}$$

From this equation, we obtain the following solution.

$$\ln q(\boldsymbol{\pi}) = \ln \Pr(\boldsymbol{\pi}; \boldsymbol{\alpha}) + \frac{1}{n} \sum_{i=1}^{n} \langle \ln \Pr(t_{i1} \mid \boldsymbol{\pi}) \rangle_{q(\boldsymbol{t}_i)} - \lambda'_\pi \qquad (11)$$

where $\lambda'_\pi = 1 - \frac{1}{n} \lambda_\pi$. This equation means that the function $q(\boldsymbol{\pi})$ is the Dirichlet distribution described in eq. (6).

Similarly, for each state s, by equating the functional derivative w.r.t. the variational posterior $q(\boldsymbol{\tau}_s)$ to 0, we obtain the following formula.

$$\begin{aligned}
\ln q(\boldsymbol{\tau}_s) &= \frac{1}{n} \sum_{i=1}^{n} \sum_{\boldsymbol{t}_i \in \mathcal{T}(\boldsymbol{w}_i)} q(\boldsymbol{t}_i) \sum_{j=1}^{|\boldsymbol{t}_i|-1} \ln \Pr(t_{ij} t_{ij+1} \mid \boldsymbol{\tau}) \\
&\quad + \ln \Pr(\boldsymbol{\tau}_s; \boldsymbol{\beta}) - \lambda'_{\tau_s} \qquad (12)
\end{aligned}$$

This equation means that the function $q(\boldsymbol{\tau}_s)$ is the Dirichlet distribution described in eq. (6).

Similarly, for each state s, by equating the functional derivative w.r.t. the variational posterior $q(\boldsymbol{o}_s)$ to 0, we obtain the following formula.

$$\begin{aligned}
\ln q(\boldsymbol{o}_s) &= \sum_{i=1}^{n} \sum_{\boldsymbol{t}_i \in \mathcal{T}(\boldsymbol{w}_i)} q(\boldsymbol{t}_i) \sum_{j=1}^{|\boldsymbol{t}_i|} \ln \Pr(t_{ij} \boldsymbol{w}_{ij} \mid \boldsymbol{o}) + \\
&\quad \ln \Pr(\boldsymbol{o}_s; \boldsymbol{\gamma}) - \lambda'_{o_s} \qquad (13)
\end{aligned}$$

This equation means that the function $q(\boldsymbol{o}_s)$ is the Dirichlet distribution described in eq. (6).

A.2 VBE step

In the VBE step, by equating the functional derivative w.r.t. $q(\boldsymbol{t}_i)$ to 0,

$$\begin{aligned}
&\frac{\partial \mathcal{L}_{VB}}{\partial q(\boldsymbol{t}_i)} \\
&= \int q(\boldsymbol{\pi}) \ln \Pr(t_{i1} \mid \boldsymbol{\pi}) d\boldsymbol{\pi} + \int q(\boldsymbol{\tau}) \sum_{j=1}^{|\boldsymbol{t}_i|-1} \ln \Pr(t_{ij} t_{ij+1} \mid \boldsymbol{\tau}) d\boldsymbol{\tau} \\
&\quad + \int q(\boldsymbol{o}) \sum_{j=1}^{|\boldsymbol{t}_i|} \ln \Pr(t_{ij} \boldsymbol{w}_{ij} \mid \boldsymbol{o}) d\boldsymbol{o} - (1 + \ln q(\boldsymbol{t}_i)) + \lambda_i \\
&= 0 \qquad (14)
\end{aligned}$$

From this formula, we derive

$$\begin{aligned}
&\ln q(\boldsymbol{t}_i) \\
&= \int q(\boldsymbol{\pi}) \ln \Pr(t_{i1} \mid \boldsymbol{\pi}) d\boldsymbol{\pi} \\
&\quad + \int q(\boldsymbol{\tau}) \sum_{j=1}^{|\boldsymbol{t}_i|-1} \ln \Pr(t_{ij} t_{ij+1} \mid \boldsymbol{\tau}) d\boldsymbol{\tau} \\
&\quad + \int q(\boldsymbol{o}) \sum_{j=1}^{|\boldsymbol{t}_i|} \ln \Pr(t_{ij} \boldsymbol{w}_{ij} \mid \boldsymbol{o}) d\boldsymbol{o} - \lambda'_i \qquad (15)
\end{aligned}$$

where $\lambda'_i \equiv \lambda_i - 1$ is constant and calculated from the condition eq. (4). Let us introduce the following parameters for initial, transition and output probabilities.

$$\begin{aligned}
\ln \tilde{\pi}_s &\equiv \int q(\boldsymbol{\pi}) \ln \pi_s d\boldsymbol{\pi} = \Psi(\alpha'_s) - \Psi\Big(\sum_{r \in S} \alpha'_r \Big) \\
\ln \tilde{\tau}_{sr} &\equiv \int q(\boldsymbol{\tau}) \ln \tau_{sr} d\boldsymbol{\tau} = \Psi(\beta'_{sr}) - \Psi\Big(\sum_{t \in S} \beta'_{st} \Big) \\
\ln \tilde{o}_{s\boldsymbol{w}} &\equiv \int q(\boldsymbol{o}) \ln o_{s\boldsymbol{w}} d\boldsymbol{o} = \Psi(\gamma'_{s\boldsymbol{w}}) - \Psi\Big(\sum_{\boldsymbol{u} \in \boldsymbol{W}_s} \gamma'_{s\boldsymbol{u}} \Big) \qquad (16)
\end{aligned}$$

Then, from eq. (15), we obtain

$$q(\boldsymbol{t}_i) \propto \tilde{\pi}_{t_{i1}} \prod_{j=1}^{|\boldsymbol{t}_i|-1} \tilde{\tau}_{t_{ij} t_{ij+1}} \prod_{j=1}^{|\boldsymbol{t}|} \tilde{o}_{t_{ij} \boldsymbol{w}_{ij}} \qquad (17)$$