EXTENSION OF DECODING PROBLEM OF HMM BASED ON L^P-NORM

Gen Hori[1,2]

1 Faculty of Business Administration, Asia University
2 Brain Science Institute, RIKEN

ABSTRACT

The decoding problem of hidden Markov model (HMM) is extended based on the L^p-norm of a vector of the log transition probabilities along the sequence of hidden states. The extended decoding problem coincides with the conventional decoding problem for $p = 1$, and with the minimax decoding problem for $p = \infty$. To solve the extended decoding problem, we introduce a family of Viterbi algorithm termed the "L^p-Viterbi algorithm" that continuously interpolates the conventional Viterbi algorithm and the minimax Viterbi algorithm. We also consider the corresponding evaluation and estimation problems. Numerical simulations show that the L^p-Viterbi algorithm with an adequately large value of p has an advantage over the minimax Viterbi algorithm.

Index Terms— Hidden Markov model (HMM), Decoding problem, Viterbi algorithm, L^p-norm

1. INTRODUCTION

Hidden Markov model (HMM) is one of the most widely implemented methods in the field of acoustics, speech and signal processing. The decoding problem of HMM is solved efficiently using the Viterbi algorithm[1] and many variants of the algorithm have been proposed so far. In the original Viterbi algorithm and most of such variants, the absolute value of the logarithm of the transition probability is considered as something like the "distance" associated with the transition and the sum of such distances along the sequence is minimized to find the optimal sequence of the hidden states, which corresponds to the maximum likelihood estimation. In some application fields, however, the absolute value of the logarithm of the transition probability should be considered as something like the height of the "hurdle" associated with the transition where it is more important to minimize the height of the tallest hurdle along the sequence than to minimize the sum of the heights of all the hurdles along the sequence. To provide a unified way of looking at both interpretations as the distance and the hurdle, we introduce a generalized decoding problem of HMM based on a cost function expressed in the form of the L^p-norm[2] of the log transition probabilities where the cases with $p = 1$ and $p = \infty$ correspond to the distance and the hurdle type applications, respectively. The generalized decoding problem can be solved efficiently using the same scheme as the Viterbi algorithm with the probability table and the back pointer table. The basic idea of the L^p-Viterbi algorithm was introduced in [3] to interpolate the conventional Viterbi algorithm and the minimax Viterbi algorithm [4] for an application of fingering decision of string instruments [5][6]. The present work extends the idea for a general case of $p > 0$, considers the corresponding evaluation and estimation problems, and shows that the L^p-Viterbi algorithm has an advantage over the minimax Viterbi algorithm through numerical simulations.

2. DECODING PROBLEM BASED ON L^P-NORM

2.1. L^p-norm

For a real number $p > 0$, the L^p-norm of an n-dimensional real vector

$$\boldsymbol{v} = (v_1, v_2, \ldots, v_n) \in R^n$$

is defined as

$$||\boldsymbol{v}||_p = \begin{cases} (|v_1|^p + |v_2|^p + \cdots + |v_n|^p)^{\frac{1}{p}} & (p \geq 1) \\ |v_1|^p + |v_2|^p + \cdots + |v_n|^p & (0 < p < 1) \end{cases}. \tag{1}$$

The upper formula can not be used for $0 < p < 1$ because it is not subadditive (does not satisfy the triangular inequality). The lower formula is subadditive for $0 < p < 1$ although it is not a norm in a strict sense ($||k\boldsymbol{v}||_p \neq k||\boldsymbol{v}||_p$) . The L^1-norm is the sum of the absolute values of the elements of $\boldsymbol{v}$,

$$||\boldsymbol{v}||_1 = |v_1| + |v_2| + \cdots + |v_n|. \tag{2}$$

The L^∞-norm and the L^0-"norm"[8] (the limit of the L^p-norm for $p \to \infty$ and $p \to 0$) are the maximum absolute value of the elements of $\boldsymbol{v}$ and the number of the nonzero elements of $\boldsymbol{v}$, respectively.

$$||\boldsymbol{v}||_\infty = \max\{|v_1|, |v_2|, \ldots, |v_n|\}, \tag{3}$$

$$||\boldsymbol{v}||_0 = |\{\ i \mid 1 \leq i \leq n,\ v_i \neq 0\}|. \tag{4}$$

2.2. Hidden Markov model (HMM)

Suppose that we have two finite sets of hidden states Q and output symbols S,

$$Q = \{q_1, q_2, \ldots, q_N\},$$
$$S = \{s_1, s_2, \ldots, s_M\},$$

and two sequences of random variables $\boldsymbol{X}$ of hidden states and $\boldsymbol{Y}$ of output symbols,

$$\begin{aligned}\boldsymbol{X} &= (X_1, X_2, \ldots, X_T), \quad X_t \in Q,\\ \boldsymbol{Y} &= (Y_1, Y_2, \ldots, Y_T), \quad Y_t \in S,\end{aligned}$$

then a hidden Markov model H is defined by a triplet

$$H = (A, B, \Pi)$$

where A is an $N \times N$ matrix of the transition probabilities,

$$A = (a_{ij}),\ a_{ij} \equiv a(q_i, q_j) \equiv P(X_t = q_j | X_{t-1} = q_i),$$

B an $N \times M$ matrix of the output probabilities,

$$B = (b_{ik}),\ b_{ik} \equiv b(q_i, s_k) \equiv P(Y_t = s_k | X_t = q_i),$$

and Π an N-dimensional vector of the initial distribution of hidden states,

$$\Pi = (\pi_i),\ \pi_i \equiv \pi(q_i) \equiv P(X_1 = q_i).$$

2.3. Conventional Viterbi algorithm

When we observe a sequence of output symbols

$$\boldsymbol{y} = (y_1, y_2, \ldots, y_T)$$

from a hidden Markov model H, we are interested in the sequence of hidden states

$$\boldsymbol{x} = (x_1, x_2, \ldots, x_T)$$

that generated the observed sequence of output symbols $\boldsymbol{y}$ with the maximum likelihood,

$$\begin{aligned}\hat{\boldsymbol{x}} &= \arg\max_{\boldsymbol{x}} P(\boldsymbol{y}, \boldsymbol{x} | H)\\ &= \arg\max_{\boldsymbol{x}} P(\boldsymbol{x}|H) P(\boldsymbol{y}|\boldsymbol{x}, H)\\ &= \arg\max_{\boldsymbol{x}} (\log P(\boldsymbol{x}|H) + \log P(\boldsymbol{y}|\boldsymbol{x}, H))\\ &= \arg\max_{\boldsymbol{x}} \sum_{t=1}^{T} (\log a(x_{t-1}, x_t) + \log b(x_t, y_t)), \qquad (5)\end{aligned}$$

where we write $\pi(x_1) = a(x_0, x_1)$ for convenience. Where we define a vector of the transition probabilities $\boldsymbol{a}(\boldsymbol{x})$ and a vector of the output probabilities $\boldsymbol{b}(\boldsymbol{x}, \boldsymbol{y})$ along the hidden sequence $\boldsymbol{x}$ and the output sequence $\boldsymbol{y}$,

$$\begin{aligned}\boldsymbol{a}(\boldsymbol{x}) &= (\pi(x_1), a(x_1, x_2), \ldots, a(x_{T-1}, x_T)),\\ \boldsymbol{b}(\boldsymbol{x}, \boldsymbol{y}) &= (b(x_1, y_1), b(x_2, y_2), \ldots, b(x_T, y_T)),\end{aligned}$$

we can rewrite (5) as,

$$\hat{\boldsymbol{x}}_{L^1} = \arg\min_{\boldsymbol{x}} || - \log \boldsymbol{a}(\boldsymbol{x}) - \log \boldsymbol{b}(\boldsymbol{x}, \boldsymbol{y}) ||_1, \qquad (6)$$

where $\log$ operates element-wise on a vector. The problem of finding the maximum likelihood sequence $\hat{\boldsymbol{x}}_{L^1}$ is called the "decoding problem" and solved efficiently using two $N \times T$ tables $\Delta = (\delta_{it})$ of log probabilities and $\Psi = (\psi_{it})$ of back pointers and the following four steps.

Initialization initializes the first columns of the two tables Δ and Ψ using the following formulae for $i = 1, 2, \ldots, N$,

$$\begin{aligned}\delta_{i1} &= -\log \pi_i - \log b(q_i, y_1),\\ \psi_{i1} &= 0.\end{aligned}$$

Recursion fills out the rest columns of Δ and Ψ using the following recursive formulae for $j = 1, 2, \ldots, N$ and $t = 1, 2, \ldots, T$–1,

$$\begin{aligned}\delta_{j,t+1} &= \min_i (\ \delta_{it} + (-\log a_{ij} - \log b(q_j, y_{t+1}))\ ),\\ \psi_{j,t+1} &= \arg\min_i (\ \delta_{it} + (-\log a_{ij} - \log b(q_j, y_{t+1}))\ ).\end{aligned}$$

Termination finds the index of the last hidden state of the maximum likelihood sequence $\hat{\boldsymbol{x}}_{L^1}$ using the last column of Δ,

$$i_T = \arg\min_i \delta_{iT}.$$

Backtracking tracks the indices of the hidden states of the maximum likelihood sequence $\hat{\boldsymbol{x}}_{L^1}$ from the last to the first using the back pointers of Ψ for $t = T, T$–$1, \ldots, 2$,

$$i_{t-1} = \psi_{i_t, t},$$

from which $\hat{\boldsymbol{x}}_{L^1}$ is obtained as

$$x_t = q_{i_t}\ (t = 1, 2, \ldots, T).$$

2.4. L^p-Viterbi algorithm

Now we extend the decoding problem of HMM (6) for an arbitrary positive value of p,

$$\hat{\boldsymbol{x}}_{L^p} = \arg\min_{\boldsymbol{x}} || - \log \boldsymbol{a}(\boldsymbol{x}) - \log \boldsymbol{b}(\boldsymbol{x}, \boldsymbol{y}) ||_p. \qquad (7)$$

According to (2) and (3), the special cases of (7) with $p = 1$ and $p = \infty$ correspond to the distance and the hurdle type applications, respectively. We call the decoding problem (7) the "L^p-decoding problem." We can solve the L^p-decoding problem efficiently by modifying the first and the second step of the conventional Viterbi algorithm as follows. We call this variant the "L^p-Viterbi algorithm."

Initialization for L^p-Viterbi algorithm initializes the first columns of the two tables Δ and Ψ using the following formulae for $i = 1, 2, \ldots, N$,

$$\begin{aligned}\delta_{i1} &= (-\log \pi_i - \log b(q_i, y_1))^p,\\ \psi_{i1} &= 0.\end{aligned}$$

Recursion for L^p-Viterbi algorithm fills out the two tables Δ and Ψ using the following recursive formulae for $j = 1, 2, \ldots, N$ and $t = 1, 2, \ldots, T$–1,

$$\delta_{j,t+1} = \min_i(\ \delta_{it} + (-\log a_{ij} - \log b(q_j, y_{t+1}))^p\ ),$$
$$\psi_{j,t+1} = \arg\min_i(\ \delta_{it} + (-\log a_{ij} - \log b(q_j, y_{t+1}))^p\ ).$$

2.5. Minimax Viterbi algorithm (L^∞-Viterbi algorithm)

According to (3), the L^p-decoding problem (7) for $p = \infty$ can be written as

$$\begin{aligned}\hat{\boldsymbol{x}}_{L^\infty} &= \arg\min_{\boldsymbol{x}} ||-\log \boldsymbol{a}(\boldsymbol{x}) - \log \boldsymbol{b}(\boldsymbol{x}, \boldsymbol{y})||_\infty \\ &= \arg\min_{\boldsymbol{x}}\ \max(-\log \boldsymbol{a}(\boldsymbol{x}) - \log \boldsymbol{b}(\boldsymbol{x}, \boldsymbol{y})) \\ &= \arg\min_{\boldsymbol{x}}\ \max_t(-\log a(x_{t-1}, x_t) - \log b(x_t, y_t)) \quad (8)\end{aligned}$$

where $\max(\boldsymbol{v})$ is the maximum element of a real vector $\boldsymbol{v}$. We call the decoding problem (8) the "minimax decoding problem." This is a problem of minimizing the maximum absolute value of the log probability associated with transition along the hidden sequence and describes the hurdle type applications. To implement a variant of the conventional Viterbi algorithm for the hurdle type applications, we modify the second step of the conventional Viterbi algorithm as follows. We call this variant the "minimax Viterbi algorithm" or the "∞-Viterbi algorithm."

Recursion for minimax Viterbi algorithm fills out the two tables Δ and Ψ using the following recursive formulae for $j = 1, 2, \ldots, N$ and $t = 1, 2, \ldots, T$–1,

$$\delta_{j,t+1} = \min_i(\max(\delta_{it},\ -\log a_{ij} - \log b(q_j, y_{t+1}))),$$
$$\psi_{j,t+1} = \arg\min_i(\max(\delta_{it},\ -\log a_{ij} - \log b(q_j, y_{t+1}))).$$

3. EVALUATION AND ESTIMATION

3.1. L^p-evaluation

In terms of probability, the L^p-decoding problem discussed in the previous section corresponds to the extension of the product of probabilities,

$$P(\boldsymbol{y}, \boldsymbol{x}|H) = \prod_{t=1}^{T} a(x_{t-1}, x_t) b(x_t, y_t), \quad (9)$$

to the exponential of the L^p-norm of log probabilities,

$$z(\boldsymbol{x}, \boldsymbol{y}, p) = e^{-||-\log \boldsymbol{a}(\boldsymbol{x}) - \log \boldsymbol{b}(\boldsymbol{x}, \boldsymbol{y})||_p}. \quad (10)$$

We denote the sum of $z(\boldsymbol{x}, \boldsymbol{y}, p)$ for all the hidden sequences $\boldsymbol{x}$ and output sequences $\boldsymbol{y}$ by $Z(p)$,

$$Z(p) = \sum_{\boldsymbol{x}\in\mathcal{X}} \sum_{\boldsymbol{y}\in\mathcal{Y}} z(\boldsymbol{x}, \boldsymbol{y}, p),$$

where $\mathcal{X}$ and $\mathcal{Y}$ are the sets of hidden sequences and output sequences, respectively ($|\mathcal{X}| = N^T$, $|\mathcal{Y}| = M^T$) . Obviously, we have $Z(1) = 1$. Then the conventional evaluation of an output sequence $\boldsymbol{y}$,

$$P(\boldsymbol{y}|H) = \sum_{\boldsymbol{x}\in\mathcal{X}} P(\boldsymbol{y}, \boldsymbol{x}|H),$$

is extended to the L^p-evaluation of $\boldsymbol{y}$,

$$\frac{1}{Z(p)} \sum_{\boldsymbol{x}\in\mathcal{X}} z(\boldsymbol{x}, \boldsymbol{y}, p). \quad (11)$$

3.2. L^p-estimation

We introduce two binary variables that represent whether a hidden sequence $\boldsymbol{x}$ stays at q_i at time t and transitions from q_i to q_j at time $t + 1$,

$$c(\boldsymbol{x}, i, t) = \begin{cases} 1 & \text{if } x_t = q_i \\ 0 & \text{otherwise} \end{cases},$$
$$c(\boldsymbol{x}, i, j, t) = c(\boldsymbol{x}, i, t)\, c(\boldsymbol{x}, j, t + 1),$$

and a set of indices at which the output symbol of $\boldsymbol{y}$ is s_k,

$$T(\boldsymbol{y}, k) = \{\, t \mid y_t = s_k \,\}.$$

Then the summations of the binary variables weighted by the exponential of the L^p-norm of log probabilities (10),

$$C(\boldsymbol{y}, i, t, p) = \sum_{\boldsymbol{x}\in\mathcal{X}} c(\boldsymbol{x}, i, t) z(\boldsymbol{x}, \boldsymbol{y}, p),$$
$$C(\boldsymbol{y}, i, j, t, p) = \sum_{\boldsymbol{x}\in\mathcal{X}} c(\boldsymbol{x}, i, j, t) z(\boldsymbol{x}, \boldsymbol{y}, p),$$

correspond to the likeliness of the hidden sequence $\boldsymbol{x}$ staying at q_i at time t and transitioning from q_i to q_j at time $t+1$ when the output sequence is $\boldsymbol{y}$. The ratios of those summations give the update rules of the EM algorithm for the L^p-estimation of the model parameters as,

$$\begin{aligned}\pi_i &\leftarrow \frac{\sum_{\boldsymbol{y}\in D} C(\boldsymbol{y}, i, 1, p)}{\sum_{\boldsymbol{y}\in D} \sum_{i=1}^{N} C(\boldsymbol{y}, i, 1, p)}, \\ a_{ij} &\leftarrow \frac{\sum_{\boldsymbol{y}\in D} \sum_{t=1}^{T-1} C(\boldsymbol{y}, i, j, t, p)}{\sum_{\boldsymbol{y}\in D} \sum_{t=1}^{T-1} C(\boldsymbol{y}, i, t, p)}, \\ b_{ik} &\leftarrow \frac{\sum_{\boldsymbol{y}\in D} \sum_{t\in T(\boldsymbol{y},k)} C(\boldsymbol{y}, i, t, p)}{\sum_{\boldsymbol{y}\in D} \sum_{t=1}^{T} C(\boldsymbol{y}, i, t, p)},\end{aligned} \quad (12)$$

where D is the set of learning data $\boldsymbol{y}$. Unfortunately, we can not make efficient recursive procedures for calculating (11) or (12) such as the forward probability or the Baum-Welch algorithm[9], mainly because the distributive property, which holds for the product of probabilities (9), does not hold for the exponential of the L^p-norm of log probabilities (10) so that we can not sum the results for the subsequences and use the sum in the rest of the calculation.

4. NUMERICAL EXPERIMENTS

We perform two numerical experiments to show that i) the L^p-Viterbi algorithm and the minimax Viterbi algorithm actually reduce the height of the tallest hurdle and ii) we should replace the minimax Viterbi algorithm with the L^p-Viterbi algorithm with an adequately large value of p for solving practical hurdle type applications. We generate toy HMM examples with 10 hidden states, 10 output symbols and randomly generated initial, transition and output probabilities for solving the decoding problem of a randomly generated output sequence with a length of 10 using the L^p-Viterbi algorithm varying the value of p. We repeat the experiment 100 times for each value of p, for each of which the initial, transition and output probabilities and the output sequence are randomly generated. Where we define a generic cost function

$$\phi_{L^p}(\boldsymbol{x}, \boldsymbol{y}) = || - \log \boldsymbol{a}(\boldsymbol{x}) - \log \boldsymbol{b}(\boldsymbol{x}, \boldsymbol{y})||_p,$$

(6), (7) and (8) can be written in a unified way as

$$\hat{\boldsymbol{x}}_{L^p} = \arg\min_{\boldsymbol{x}} \phi_{L^p}(\boldsymbol{x}, \boldsymbol{y}).$$

Figure 1 shows the average values of the two cost functions $\phi_{L^\infty}(\hat{\boldsymbol{x}}_{L^p}, \boldsymbol{y})$, the height of the tallest hurdle, and $\phi_{L^1}(\hat{\boldsymbol{x}}_{L^p}, \boldsymbol{y})$, the sum of the heights of all the hurdles, from which we see that the L^p-Viterbi algorithm actually decreases the height of the tallest hurdle on the cost of increasing the sum of the heights of all the hurdles. Figure 2 shows the scatter plots of the two cost functions $\phi_{L^\infty}(\hat{\boldsymbol{x}}_{L^p}, \boldsymbol{y})$ and $\phi_{L^1}(\hat{\boldsymbol{x}}_{L^p}, \boldsymbol{y})$. In most of practical hurdle type applications, it is preferable to choose a hidden sequence with the minimum height of the tallest hurdle whose other hurdles are as low as possible. From the two bottom panels of Figure 2, we see that the L^p-Viterbi algorithm with an adequately large value of p almost always finds a hidden sequence with the same value of the minimum height of the tallest hurdle as the minimax Viterbi algorithm and significantly lowers the sum of the heights of all the hurdles.

5. CONCLUSION

We have introduced a variant of the conventional Viterbi algorithm termed the "L^p-Viterbi algorithm" that finds a sequence of hidden states that minimizes the L^p-norm of a vector of the log transition probabilities. It has been shown that the extended decoding problem defined with the L^p-norm can be solved efficiently using the L^p-Viterbi algorithm whereas the corresponding evaluation and estimation problems can not be solved efficiently because the distributive property does not hold for the exponential of the L^p-norm of log probabilities. Our numerical experiments have shown that the L^p-Viterbi algorithm with an adequately large value of p has an advantage over the minimax Viterbi algorithm when applied to practical hurdle type applications.

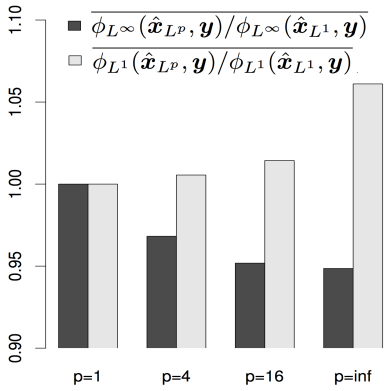

Fig. 1. The average values of $\phi_{L^\infty}(\hat{\boldsymbol{x}}_{L^p}, \boldsymbol{y})$ (dark gray) and $\phi_{L^1}(\hat{\boldsymbol{x}}_{L^p}, \boldsymbol{y})$ (light gray) of the L^p ($p = 1, 4, 16$) and the minimax ($p = \infty$) Viterbi paths relative to the L^1 (conventional) Viterbi path ($p = 1$). We see that, as p increases, the average height of the tallest hurdles actually decreases while the sum of the heights of all the hurdles increases.

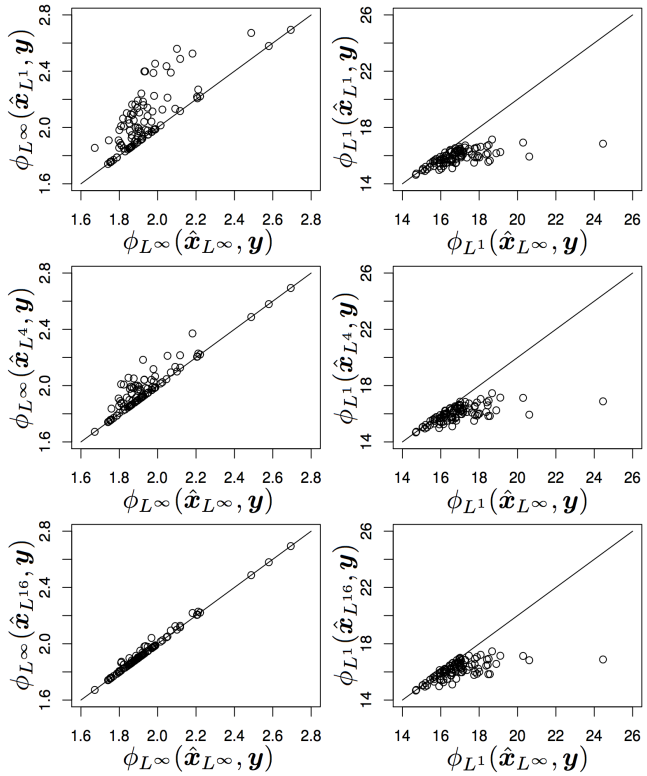

Fig. 2. The scatter plots of $\phi_{L^\infty}(\hat{\boldsymbol{x}}_{L^p}, \boldsymbol{y})$ (left panels) and $\phi_{L^1}(\hat{\boldsymbol{x}}_{L^p}, \boldsymbol{y})$ (right panels) of the L^p ($p = 1, 4, 16$, vertical axis) versus the minimax ($p = \infty$, horizontal axis) Viterbi paths. We see from the bottom panels that, for an adequately large value of $p = 16$, the L^p-Viterbi algorithm almost always attains the same minimum height of the tallest hurdle as the minimax Viterbi algorithm (bottom left) and significantly lowers the sum of the heights of all the hurdles comparing to the minimax Viterbi algorithm (bottom right).

6. REFERENCES

[1] Andrew J Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Transactions on Information Theory*, vol. 13, no. 2, pp. 260–269, 1967.

[2] Jöran Bergh and Jörgen Löfström, "Interpolation of L^p-spaces," in *Interpolation Spaces*, pp. 106–130. Springer, 1976.

[3] Gen Hori and Shigeki Sagayama, "L^p-Viterbi algorithm for automatic fingering decision," in *Proceedings of the 14th Sound and Music Computing Conference (SMC2017)*, Espoo, Finland, 2017, pp. 386–390.

[4] Gen Hori and Shigeki Sagayama, "Minimax Viterbi algorithm for HMM-based guitar fingering decision," in *Proceedings of 17th International Society for Music Information Retrieval (ISMIR2016)*, New York City, U.S.A., 2016, pp. 448–453.

[5] Gen Hori, Hirokazu Kameoka, and Shigeki Sagayama, "Input-output HMM applied to automatic arrangement for guitars," *Journal of Information Processing*, vol. 21, no. 2, pp. 264–271, 2013.

[6] Samir I Sayegh, "Fingering for string instruments with the optimum path paradigm," *Computer Music Journal*, vol. 13, no. 3, pp. 76–84, 1989.

[7] Yoshua Bengio and Paolo Frasconi, "An input output HMM architecture," *Advances in neural information processing systems*, vol. 7, pp. 427–434, 1995.

[8] David L Donoho, "For most large underdetermined systems of linear equations the minimal l^1-norm solution is also the sparsest solution," *Communications on Pure and Applied Mathematics*, vol. 59, no. 6, pp. 797–829, 2006.

[9] Leonard E Baum and Ted Petrie, "Statistical inference for probabilistic functions of finite state Markov chains," *The annals of mathematical statistics*, vol. 37, no. 6, pp. 1554–1563, 1966.