CONDITIONAL ITERATIVE DECODING OF TWO DIMENSIONAL HIDDEN MARKOV MODELS

M. E. Sargin, A. Altinok, K. Rose, B. S. Manjunath

Department of Electrical and Computer Engineering,
University of California Santa Barbara, Santa Barbara, CA 93106
{msargin,alphan,rose,manj}@ece.ucsb.edu

ABSTRACT

Two Dimensional Hidden Markov Models (2D-HMMs) provide substantial benefits for many computer vision and image analysis applications. Many fundamental image analysis problems, including segmentation and classification, are target applications for the 2D-HMMs. As opposed to the i.i.d. assumption of the image observations, the naturally existing spatial correlations can be readily modeled by solving the 2D-HMM decoding problem. However, computational complexity of the 2D-HMM decoding grows exponentially with the image size and is known to be NP-hard. In this paper, we present a Conditional Iterative Decoding (CID) algorithm for the approximate decoding of 2D-HMMs. We compare the performance of the CID algorithm to the Turbo-HMM (T-HMM) decoding algorithm and show that CID gives promising results. We demonstrate the proposed algorithm on modeling spatial deformations of human faces in recognizing people across their different facial expressions.

Index Terms— Image analysis, Hidden Markov Models

1. INTRODUCTION

Conventional HMMs (1D-HMMs) have been successfully used in modeling temporal dependencies of numerous Markovian processes. Main advantage of 1D-HMM on time series data is the existence of computationally efficient algorithms for both learning the model parameters (training) and finding the optimal state sequence given the data and the model (decoding).

The 2D extension of the 1D-HMM, named 2D-HMM, has been applied in [1, 2, 3, 4] to model naturally existing spatial correlations on images. Unfortunately, if we were to adopt the training and decoding algorithms from 1D-HMM and use them in the 2D-HMM context, the computational complexity grows exponentially with the image size and hence the problem becomes intractable. Several approximation algorithms are proposed to reduce the complexity and make the problem tractable.

Path Constrained Variable State Viterbi (PCVSV) algorithm, [1], reduces the computational complexity by limiting the Viterbi search space. Only K state sequences with highest observation probabilities are considered, without explicitly calculating their posteriors. Since the PCVSV may converge to a local solution, one must select K large enough to obtain a "good" solution, considering the size of the image. On the other hand, to control the complexity, K must be limited or the image must be analyzed in non-overlapping blocks ignoring the correlations between pixels on the borders of blocks.

This study was funded by Center for Bioimage Informatics under grants NSF-ITR 0331697.

More recently, an iterative decoding algorithm for the 2D-HMM was presented as Turbo HMM (T-HMM), [5, 6]. Here, the authors propose to apply 1D-decoding on rows and columns independent of each other. Then, the posterior state probabilities of rows (columns) are used in decoding the columns (rows) as prior probabilities. Thus, the horizontal and vertical processes "communicate" through the posterior state probabilities. The idea is a row- and column-wise constrained application of belief propagation, [5, 6]. The main assumption is to represent the dependency from the neighbors as the horizontal and vertical conditionals. This *separation* of horizontal and vertical dependencies is too restrictive for a generic 2D-HMM decoding task.

In this paper, we describe a conditional iterative decoding (CID) algorithm for decoding 2D-HMMs. In our algorithm we do not assume that the transition matrix can be separated into its horizontal and vertical components, thus the performance improves upon [7] in the general 2D-HMM decoding problem. The proposed method consists of *ordered* iterative updates on rows and columns. Instead of decoding the rows (columns) independently, we decode the rows (columns) using the posteriors from the previous row (column) *and* the posteriors of the corresponding column (row) calculated in the previous iteration.

The rest of the paper is organized as follows. In Section 2, we introduce the notation used throughout the paper and explain the method for exact decoding of the 2D-HMM with its computational complexity. Then, we describe our proposed conditional iterative decoding (CID) algorithm in Section 3. We finally present experimental results and give concluding remarks on Section 4 and Section 5 respectively.

2. DECODING OF THE 2D-HMM

Consider a set of nodes on a 2D lattice of size $I \times J$. Each node at (i, j) corresponds to a hidden state, $q_{i,j}$ of the 2D-HMM. Let S_n denote the possible values that the state $q_{i,j}$ can take, Fig.1. We assume that the probability of the state $q_{i,j}$ taking the value S_n, given all of the previous (spatially) states $q_{i',j'}$ where $(i', j') \in \{(i', j') : i' < i \text{ or } j' < j\}$, can be written as

$$P(q_{i,j}|q_{i',j'}) = P(q_{i,j}|q_{i-1,j}, q_{i,j-1}). \quad (1)$$

Accordingly, we define the 3D transition matrix $\mathbf{A} = [a_{k,m,n}]$ as $a_{k,m,n} = P(q_{i,j} = S_n|q_{i-1,j} = S_k, q_{i,j-1} = S_m)$. The observation at $q_{i,j}$ is denoted by $o_{i,j}$ and the probability of observing $o_{i,j}$ depends only on the value of the state $q_{i,j}$. The observation distributions are represented with $b_n(o_{i,j}) = P(o_{i,j}|q_{i,j} = S_n)$, and the set of all observation distributions $\{b_n\}$ is $\mathbf{B}$.

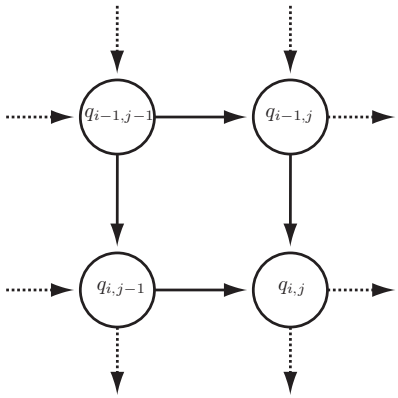

Fig. 1. Bayesian network of the 2D-HMM.

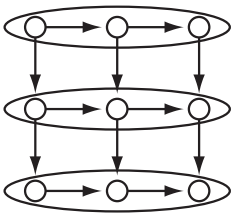

Fig. 2. The state sequences of the equivalent 1D-HMM

Recall that in [7], the transition matrix was decomposed into its vertical and horizontal components, call these $\mathbf{A}^v = [a^v_{k,n}]$ and $\mathbf{A}^h = [a^h_{m,n}]$ with $a^v_{k,n} = P(q_{i,1} = S_n | q_{i-1,1} = S_k)$ and $a^h_{m,n} = P(q_{1,j} = S_n | q_{1,j-1} = S_m)$. Here, we use this decomposition for our initial row and column computation. Accordingly, when $(i = 1, j = 1)$, we are only left with $P(q_{1,1} = S_n) = \pi_n$ where $\pi = [\pi_n]$ is the prior probabilities for $q_{1,1}$.

The set of all observations is represented with $\mathbf{O} = \{o_{i,j} : i \in \mathcal{I}, j \in \mathcal{J}\}$ and the set of observations from i^{th} row and j^{th} column are represented with $\mathbf{o}^h_i = \{o_{i,j} : j \in \mathcal{J}\}$ and $\mathbf{o}^v_j = \{o_{i,j} : i \in \mathcal{I}\}$ respectively. $\mathbf{Q}$, $\mathbf{q}^h_i$ and $\mathbf{q}^v_j$ are also defined similarly based on $q_{i,j}$'s.

By decoding, we refer to finding the best state sequence $\mathbf{Q}^*$ given the observations $\mathbf{O}$ and the model λ such that:

$$\mathbf{Q}^* = \underset{\mathbf{Q}}{\operatorname{argmax}}\, P(\mathbf{Q}|\mathbf{O}, \lambda) = \underset{\mathbf{Q}}{\operatorname{argmax}}\, P(\mathbf{O}, \mathbf{Q}|\lambda). \qquad (2)$$

A 2D-HMM can be converted into an equivalent 1D-HMM. In this case, each state sequence of the nodes that are enclosed with ellipses corresponds to a single state in the equivalent 1D-HMM, Fig 2. Then, the decoding could be performed by the Viterbi algorithm. However, the number of states needed to represent all state sequences of the corresponding 2D-HMM would grow exponentially ($N^{\min(I,J)}$). Therefore, the exact decoding of a 2D-HMM is an NP-hard problem.

3. PROPOSED ALGORITHM

The algorithm consists of conditional iterative updates of the posteriors on rows and columns. The method described in [7] assumes the *separability* of the transition matrix into two matrices representing row and column transitions. Thus, the vertical and horizontal dependencies are calculated independently. Instead, we propose to use the

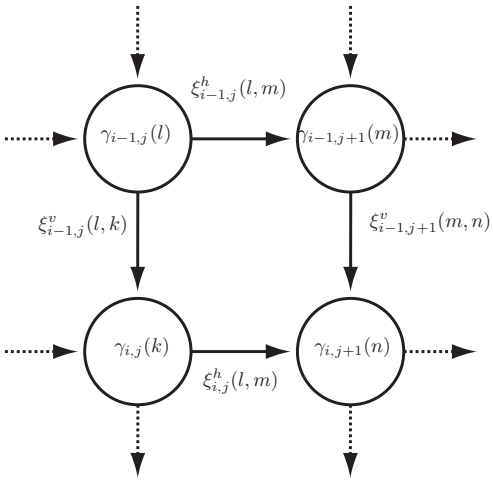

Fig. 3. Posterior probabilities of the 2D-HMM. Note that, ξ's associated with directed edges are joint probability distributions.

posteriors from the previous row and column to calculate the next row and column. In other words, posterior probabilities extracted from each scan is conditioned on the previous scan along rows and columns. While we use the row and column decomposition in the initial scan, we perform the subsequent updates over the entire 3D transition matrix as opposed to the vertical and horizontal transition matrices. This enables us to pass the beliefs from each scan to the next one through the whole transition matrix.

Let $\gamma^h_{i,j}(n)$ and $\gamma^v_{i,j}(n)$ be the posterior probabilities of the state $q_{i,j}$ being S_n after scanning i^{th} row and j^{th} column respectively.

$$\gamma^h_{i,j}(n) = \frac{\alpha^h_{i,j}(n)\beta^h_{i,j}(n)}{\sum_n \alpha^h_{i,j}(n)\beta^h_{i,j}(n)} \qquad (3)$$

$\xi^h_{i,j}(k,n)$ represents the joint posterior probability of the states $q_{i,j}$ and $q_{i,j+1}$ being S_k and S_n respectively after scanning i^{th} row. Similarly, $\xi^v_{i,j}(m,n)$ represents the posterior probability of the states $q_{i,j}$ and $q_{i+1,j}$ being S_m and S_n respectively after scanning j^{th} column.

$$\xi^h_{i,j}(k,n) = \frac{\alpha^h_{i,j}(k)a_{i,j}(k,n)b^h_k(o_{i,j+1})\beta^h_{i,j+1}(n)}{\sum_{k,n} \alpha^h_{i,j}(k)a_{i,j}(k,n)b^h_k(o_{i,j+1})\beta^h_{i,j+1}(n)} \qquad (4)$$

where

$$a_{i,j}(k,n) = \frac{1}{Z_k} \sum_{l,m} \frac{\xi^v_{i-1,j}(l,k)\xi^h_{i-1,j}(l,m)a_{k,m,n}}{\sum_s \xi^v_{i-1,j}(l,s)} \qquad (5)$$

$$Z_k = \sum_n \sum_{l,m} \frac{\xi^v_{i-1,j}(l,k)\xi^h_{i-1,j}(l,m)a_{k,m,n}}{\sum_s \xi^v_{i-1,j}(l,s)} \qquad (6)$$

In the proposed algorithm, horizontal and vertical scans applied successively and the processes *communicate* through both γ's and ξ's. $\gamma^v_{i,j}(n)$ and $\gamma^h_{i,j}(n)$ are used to weight observation probabilities, $b_n(o_{i,j})$, to have $b^h_n(o_{i,j})$ and $b^v_n(o_{i,j})$ respectively. $\xi^v_{i-1,j}$ is incorporated to determine the transition matrix between the states

$q_{i,j}$ and $q_{i,j+1}$ during the horizontal scan. Similarly, $\xi^h_{i,j-1}$ is incorporated to determine the transition matrix between states $q_{i,j}$ and $q_{i+1,j}$ during the vertical scan.

$$b^h_n(o_{i,j}) = \gamma^v_{i,j}(n) b_n(o_{i,j}) \tag{7}$$

We obtain Equation 5 by approximating joint density of $q_{i,j}$, $q_{i,j+1}$, $q_{i-1,j}$ and $q_{i-1,j+1}$ as

$$P(q_{i,j},q_{i,j+1},q_{i-1,j},q_{i-1,j+1}) \approx P(q_{i-1,j},q_{i-1,j+1}) \\ P(q_{i,j}|q_{i-1,j})P(q_{i,j+1}|q_{i,j},q_{i-1,j}). \tag{8}$$

and then approximately marginalizing $P(q_{i,j}, q_{i,j+1}, q_{i-1,j}, q_{i-1,j+1})$ as

$$P(q_{i,j},q_{i,j+1}) \approx \sum_{q_{i-1,j}} \sum_{q_{i-1,j+1}} P(q_{i-1,j}, q_{i-1,j+1}|\mathbf{o}^h_{i-1}) \\ P(q_{i,j}|q_{i-1,j},\mathbf{o}^v_j)P(q_{i,j+1}|q_{i,j},q_{i-1,j}). \tag{9}$$

In the following section we provide forward and backward update rules for the horizontal scan. Similar formulation can be derived for vertical scan.

3.0.1. Horizontal Forward Iterations

- Initialization ($j = 1$):

$$\alpha^h_{i,j}(n) = \pi_i(n) b^h_n(o_{i,j})$$

- Induction ($j = 2, \ldots, J$):

$$\alpha^h_{i,j}(n) = b^h_n(o_{i,j}) \sum_k \alpha^h_{i,j-1}(k) a_{i,j-1}(k,n)$$

3.0.2. Horizontal Backward Iterations

- Initialization ($j = J$):

$$\beta^h_{i,j}(n) = 1$$

- Induction ($j = J-1, \ldots, 1$):

$$\beta^h_{i,j}(n) = \sum_k \beta^h_{i,j+1}(k) b^h_k(o_{i,j+1}) a_{i,j}(n,k)$$

4. EXPERIMENTAL RESULTS

4.1. Decoding Performance on Synthetic Data

Decoding performance is measured on synthetic data by comparing $\log(P(\mathbf{Q}^*|\mathbf{O},\lambda))$ for PCVSV, T-HMM and CID decoding algorithms. During the simulations, a 2D-HMM is constructed using a randomly generated transition matrix where the number of states is selected as $N = 2$. A 2D sequence of states $\mathbf{Q}$ with $I = J = 100$ is generated based on the transition matrix and the observations are obtained by adding white Gaussian noise with zero mean and $\sigma = 0.5$.

$$o_{i,j} = \begin{cases} \mathcal{N}(0,\sigma) & q_{i,j} = S_1 \\ 1 + \mathcal{N}(0,\sigma) & q_{i,j} = S_2 \end{cases} \tag{10}$$

As suggested in [7] horizontal and vertical transition matrices ($a^v_{k,n}$, $a^h_{m,n}$) are obtained by

$$a^v_{k,n} = \frac{1}{N} \sum_m a_{k,m,n}, \quad a^h_{m,n} = \frac{1}{N} \sum_k a_{k,m,n}. \tag{11}$$

The average log-likelihoods $\log(P(\mathbf{Q}^*|\mathbf{O},\lambda))/(IJ)$ with respect to the iterations are illustrated in Fig. 4. Since PCVSV is not an iterative algorithm, the average log-likelihoods are illustrated with horizontal lines with various K's.

Fig. 4 shows that, CID outperforms T-HMM and PCVSV with a reasonable K values. It is worth to mention that the log-likelihood will increase by increasing K and will reach to the exact decoding performance when $K = N^{\min(I,J)}$. However, in this case, the complexity of the PCVSV decoding will be too high to be practical. To give better idea about the complexity, for $I = J = 100$, $N = 2$ and $K = 128$, running time of T-HMM and CID were 27 times and 23 times faster than PCVSV respectively.

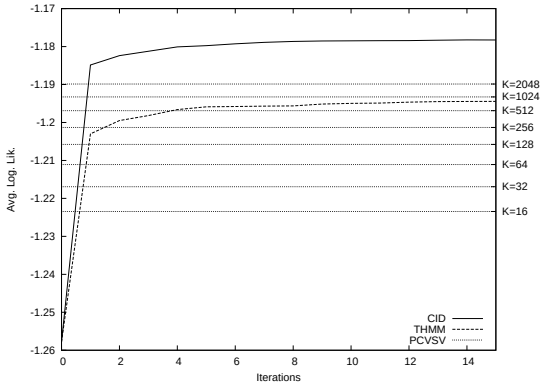

Fig. 4. Average Log-Likelihoods for CID, T-HMM and PCVSV

T-HMM assumes that the transition matrix is separable i.e., $a_{k,m,n}$ can be written as multiplication of $a^h_{m,n}$ and $a^v_{k,n}$, Eq. 11. Here, we define the measure of separability $\mathcal{D}$ as $\mathcal{D} = \sum_{k,m} \mathcal{D}_{k,m}$ where

$$D_{k,m} = \sum_n a_{k,m,n} \log(\frac{Z_{k,m} a_{k,m,n}}{a^v_{k,n} a^h_{m,n}}). \tag{12}$$

where $Z_{k,m}$ is a normalization factor such that

$$Z_{k,m} = \sum_n a^v_{k,n} a^h_{m,n}. \tag{13}$$

We run the same simulation illustrated in Fig. 4 with randomly generated transition matrices 500 times. For each simulation, we note the *separability* measure together with the difference of the average log-likelihoods of CID and T-HMM decoding algorithms.

Fig. 5 illustrates the scatter plot of the average log-likelihood differences as a function of $\mathcal{D}$. Each point indicates the difference of the average log-likelihood of CID minus that of T-HMM. Fig. 5 shows that the log-likelihood difference, hence the performance gain, improves as the $\mathcal{D}$ increases. In other words, the performance improvement of the CID over T-HMM becomes more and more significant when the *separability* assumption of the transition matrix does not hold. In addition, if the transition matrix is *separable* ($\mathcal{D} \approxeq 0$), CID does not degrade the performance over T-HMM, e.g. there are no negative values of the difference.

4.2. Decoding Performance on Deformable Face Recognition

The decoding performances of CID and T-HMM algorithms are also tested on real images on face recognition problem. We have used the Yale Face database [8], which contains 8 different facial expressions (no-glasses, surprised, glasses, sad, happy, sleepy, normal, wink)

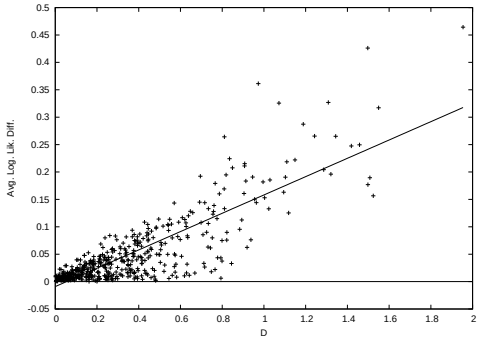

Fig. 5. Differences of Average Log-Likelihoods w.r.t $\mathcal{D}$

and 3 different illuminations (center-light, right-light, left-light) of 15 subjects. The images are cropped around the face using the coordinates of eyes and tip of nose. For each subject $s \in \{1, \ldots, 15\}$ a single image with *normal* facial expression B_s is used as the template image. Given a query image A, distance between the query and the template image is calculated based on the optimal warping of the template image onto the query image.

$$\mathbf{Q}_s^* = \underset{\mathbf{Q}}{\operatorname{argmax}}\, P(\mathbf{Q}|\mathbf{O}_{(A,B_s)}, \lambda) \tag{14}$$

In the deformable face recognition scenario, each state is associated with a translation vector and transition matrix defines the correlation of the neighboring translation vectors. Gabor wavelet features with 4 scales and 6 orientations are used to extract 24 dimensional feature vector for each pixel. The emission probabilities are extracted based on the distance of the feature vectors from image A and B_s considering the translation vectors.

Figure 6 illustrates the optimal warping of two template images with *normal* condition to the query image with *sleepy* condition. First row mesh images illustrates the optimally warped regular meshes superimposed on the template images. Second row images are the resulting warped image. It is also obvious from the figure that the cost of warping the right template image to the query image is larger than that of left template image. Most of the deformations on the mesh of the left template image tries to *close* the eyes of the template image of the subject which can also be seen on the warped image.

Once the optimal warping $\mathbf{Q}_s^*$ is found for each subject s, recognition is done based on the following:

$$s^* = \underset{s \in \{1,\ldots,15\}}{\operatorname{argmax}}\; P(\mathbf{Q}_s^*|\mathbf{O}_{(A,B_s)}, \lambda). \tag{15}$$

The overall recognition performances on 150 query images are presented on Table 1.

	T-HMM	CID
%ER	4.0	3.3

Table 1. Error Rates on Yale Face Database

5. CONCLUDING REMARKS

In this paper we present the CID algorithm for 2D-HMM decoding. In the synthetic decoding problem, we showed that performance

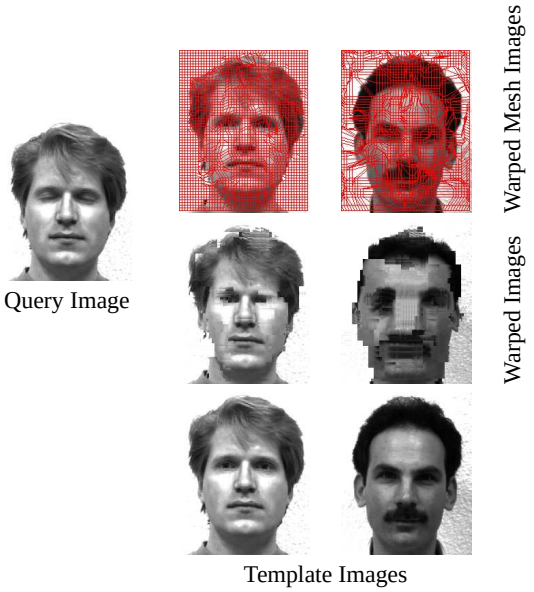

Fig. 6. Sample Face Deformation Output

improvement of CID over T-HMM increases when the *separability* measure of the transition matrix decreases and CID never degrades the performance over T-HMM even with small $\mathcal{D}$. CID also outperforms the PCVSV algorithm with reasonable K values. However, as an expected result, increasing K will always improve the performance of the PCVSV algorithm and as we reach $K = N^{\min(I,J)}$, it will be identical to the exact decoding while making the problem intractable even for small (I, J, N).

6. REFERENCES

[1] J. Li, A. Najmi, and R.M. Gray, "Image classification by a two-dimensional hidden Markov model," *Signal Processing, IEEE Transactions on*, vol. 48, no. 2, pp. 517–533, 2000.

[2] S. Kuo and OE Agazzi, "Keyword spotting in poorly printed texts using pseudo 2D hidden Markov models," *IEEE TPAMI*, vol. 16, no. 8, pp. 842–848, 1994.

[3] F. Perronnin, J.L. Dugelay, and K. Rose, "A probabilistic model of face mapping with local transformations and its application to person recognition," *IEEE TPAMI*, vol. 27, no. 7, pp. 1157–1171, 2005.

[4] H. Othman and T. Aboulnasr, "A separable low complexity 2 D HMM with application to face recognition," *IEEE TPAMI*, vol. 25, no. 10, pp. 1229–1238, 2003.

[5] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann, 1988.

[6] J.S. Yedidia, W.T. Freeman, MA Cambridge, and Y. Weiss, "Understanding Belief Propagation and Its Generalizations," *Exploring Artificial Intelligence in the New Millennium*, 2003.

[7] F. Perronnin, J.L. Dugelay, and K. Rose, "Iterative decoding of two-dimensional hidden Markov models," *Proc. of the IEEE ICASSP*, vol. 3, 2003.

[8] Yale Univ. Face Database, available at `http://cvc.yale.edu/projects/yalefaces/yalefaces.html`.