PARAMETER ESTIMATION OF MULTI-DIMENSIONAL HIDDEN MARKOV MODELS - A SCALABLE APPROACH

Dhiraj Joshi, Jia Li, and James Z. Wang

The Pennsylvania State University, University Park, PA, USA

ABSTRACT

Parameter estimation is a key computational issue in all statistical image modeling techniques. In this paper, we explore a computationally efficient parameter estimation algorithm for multi-dimensional hidden Markov models. 2-D HMM has been applied to supervised aerial image classification and comparisons have been made with the first proposed estimation algorithm. An extensive parametric study has been performed with 3-D HMM and the scalability of the estimation algorithm has been discussed. Results show the great applicability of the explored algorithm to multi-dimensional HMM based image modeling applications.

1. INTRODUCTION

Computer based content analysis of digital images has become very important today. Significant advances have been made in building efficient algorithms for intelligent search and retrieval of images from digital libraries. Learning based annotation of images, using computers, has also become possible [5]. Moreover, interest in multi-dimensional image processing has arisen from the availability of highly sensitive imaging instruments. Volumes of 3-D data is produced by MRI and CT scanners, to be used for medical analysis. Images from modern telescopes contain important information about our celestial neighbors. Several meteorological predictions are based on aerial images obtained from satellites. Hyperspectal imaging is being widely used for air surveillance and reconnaissance. A number of modern image analysis applications are built upon statistical models and require heavy computations. Thus, a critical need is to constantly strive to make these computational algorithms faster and more efficient.

In the past, one-dimensional hidden Markov models have been extended to pseudo 2-D and pseudo 3-D HMMs, for face recognition [1,2]. Two-dimensional hidden Markov models (2-D HMMs) and their multiresolution extension have been applied to supervised image classification [3, 4]. 3-D HMMs were proposed as extensions of 2-D HMMs and applied to volume image segmentation [6]. Here, we modify the parameter estimation algorithm proposed for 3-D HMMs in order to use it for 2-D HMMs and discuss the computational efficiency.

In Section 2, we state the assumptions of 2-D and 3-D hidden Markov models. Section 3 describes the parameter estimation algorithm. In Sections 4 and 5, we describe our experiments and results. Section 6 states the conclusions and future research directions.

2. MULTI-DIMENSIONAL HIDDEN MARKOV MODELS

Conventionally, 2-D images are divided into square blocks and block based features are extracted. Each block represents a point in a 2-D grid and the features extracted from block at (i, j) are represented by $u_{i,j}$. The observation $u_{i,j}$ is usually a vector containing features computed from the pixel values in the block at (i, j). A lexicographic ordering of points (i, j) is defined as follows: $(i^{'}, j^{'}) < (i, j)$ if $i^{'} < i$ or $i^{'} = i, j^{'} < j$. Under the assumptions of the 2-D HMM, the underlying states $s_{i,j}$ are governed by a first order Markov mesh: $P(s_{i,j}|s_{i',j'} : (i', j') < (i, j)) = a_{m,n,l}$, where $m = s_{i-1,j}$, $n = s_{i,j-1}$ and $l = s_{i,j}$. That is, among the conditioned states, only states of the two neighbors above and to the left of the current position affect the transition probability. Besides this, given the state $s_{i,j}$ of a point (i, j), the feature vector $u_{i,j}$ is assumed to follow a multivariate Gaussian distribution with a covariance matrix and a mean vector, determined only by its state.

Under assumptions of 3-D HMM, the states $s_{i,j,k}$ of a point (i, j, k) in a 3-D grid, are affected by three neighbors of point (i, j, k) as follows:

This work is supported by the US National Science Foundation under Grant Nos. IIS-0219272, IIS-0347148, and ANI-0202007, The Pennsylvania State University, the PNC Foundation, and SUN Microsystems under grants EDUD-7824-010456-US and EDUD-7824-030469-US.

Dhiraj Joshi is with the Department of Computer Science and Engineering, Email: djoshi@cse.psu.edu. Jia Li is with the Department of Statistics and the Department of Computer Science and Engineering, Email: jiali@stat.psu.edu. James Z. Wang is with the School of Information Sciences and Technology and the Department of Computer Science and Engineering, Email: jwang@ist.psu.edu.

$P(s_{i,j,k}|s_{i',j',k'} : (i',j',k') < (i,j,k)) = a_{p,m,n,l}$, where $p = s_{i,j,k-1}$, $m = s_{i-1,j,k}$, $n = s_{i,j-1,k}$, $l = s_{i,j,k}$, and the lexicographic ordering here is a natural extension of the ordering for 2-D grid. As in 2-D HMM case, the feature vector $u_{i,j,k}$ follows a multivariate Gaussian distribution dependent only upon the state of point (i,j,k). For details of 3-D HMM, readers are referred to [6].

3. PARAMETER ESTIMATION

For a 2-D HMM, the parameters to be estimated consist of the transition probabilities $a_{m,n,l}$, $m,n = 1 \ldots M$ where M represents the number of states, and the mean vectors μ_l, and the covariance matrices σ_l. An iterative approach is adopted. The MAP sequence of states, conditioned on the observed vectors and present parameter set is found. The parameters are then updated as maximum likelihood estimates from the data. The procedure is elaborated below.

Define the set of points with a fixed Y coordinate in a 2-D grid as a row denoted by $R_j = \{(i,j) : 0 \leq i \leq (w-1)\}$. Let $\mathcal{D} = \{j : 0 \leq j \leq w-1\}$. Denote the sequence of states and observed vectors in row R_j by $\mathbf{s}_j$ and $\mathbf{u}_j$ respectively. Rows are processed in the lexicographic order and 1-D Viterbi is used to compute the MAP sequence of states at step 4. In the following algorithm, we denote the states (in row R_j) obtained at iteration t by $\mathbf{s}_j^t$.

1. Initialize $t \leftarrow 0$, $j \leftarrow 0$.
2. If $(t = 0)$ $\mathcal{S}_j = \{\mathbf{s}_{j'}^t, j' < j\}$.
   $\mathcal{F}_j = \{\mathbf{u}_{j'}, j' \leq j\}$.
   If $(t > 0)$
   $\mathcal{S}_j = \{\mathbf{s}_{j'}^t, j' < j\} \cup \{\mathbf{s}_{j'}^{(t-1)}, j' > j\}$.
   $\mathcal{F}_j = \{\mathbf{u}_{j'}, j' \in \mathcal{D}\}$.
3. Search for $\mathbf{s}_j^*$ with MAP conditioned on $\mathcal{S}_j \cup \mathcal{F}_j$.
4. $\mathbf{s}_j^t \leftarrow \mathbf{s}_j^*$, $j \leftarrow j+1$.
5. If $j < w$, go back to step 3. Otherwise,
   (a) $t \leftarrow t+1$.
   (b) If *stopping criteria satisfied*, stop, else go to Step 2.

We stop the iterations when the number of points whose states are altered in that iteration falls below a predetermined threshold.

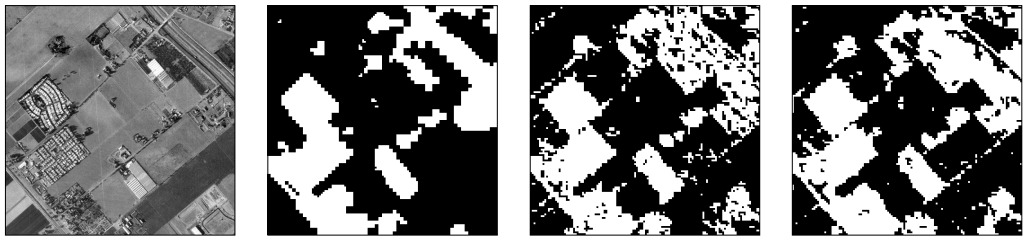
In order to search the MAP sequence of states in a row of w points, using 1-D Viterbi, the computation required is of the order of wM^2. For a 2-D grid containing $w \times w$ points (or rather w rows), one iteration of the proposed algorithm through the grid will take $\mathcal{O}(w^2M^2)$ time. A 3-D grid containing $w \times w \times w$ points has a total of w^2 rows. Therefore, the complexity of one iteration of this algorithm through the 3-D grid is $\mathcal{O}(w^3M^2)$. Thus, we see that the computational complexity of the iterative approach is *polynomial* in number of states and linear in problem size for both 2-D HMM and 3-D HMM.

Let us now discuss the parameter estimation algorithm which was proposed for 2-D HMMs in [3]. It was proposed that the MAP sequence of states in a 2-D image, represented as a $w \times w$ grid, could be obtained by considering sequence of states along diagonals of the grid and using Viterbi algorithm. If Viterbi algorithm is used to search for optimal state sequences, then the complexity would be $\mathcal{O}(M^\nu)$ where ν is the number of points in the largest diagonal. To decrease the computational complexity, a path constrained Viterbi approach was adopted, which restricted the search for state sequences at each diagonal to N sequences with largest posterior probabilities. To model an image, it was broken down into square blocks of size $k \times k$ and each block was modeled as a separate 2-D HMM. Thus, for a 2-D image represented as a $w \times w$ grid, w^2/k^2 separate 2-D HMMs were estimated. The value of the block size k was generally selected as 8. For more details of this algorithm, readers are referred to [3]. Path constrained Viterbi, for each $k \times k$ block, can be performed in $\mathcal{O}(kN^2)$ time and so the computational complexity for the entire 2-D image is roughly $\mathcal{O}(w^2N^2/k)$. Besides this, the described approach is not readily extendible to 3-D HMMs. For the remainder of the paper, we will refer to the estimation algorithm based on variable state Viterbi as *Algorithm* $\mathcal{A}$ and the estimation algorithm proposed in this paper as *Algorithm* $\mathcal{B}$.

4. SUPERVISED CLASSIFICATION OF AERIAL IMAGES USING 2-D HMM

Our experiments with 2-D aerial images were focused on comparing the proposed estimation algorithm ($\mathcal{B}$) to the algorithm based on variable state Viterbi ($\mathcal{A}$). The goal of the experiments was classification of aerial images into man-made and natural regions. The experimental setup was similar to the setup in [3]. Although classification results, for algorithm $\mathcal{A}$, are available in [3], we performed all the experiments again in order to correctly compare the learning and classification times of the two estimation programs, when run on similar processors.

The 6 images used are 512×512 gray-scale images with 8 bits per-pixel (bpp). The images are the aerial images of the San Francisco Bay area. The availability of hand-labeled classification for the 6 images proved helpful in evaluating the performances. The hand labeled classification was used as the truth set, for our experiments. The images were divided into 4×4 blocks, and DCT coefficients or averages over them were calculated and used as features. If the DCT coefficients for a 4×4 block are



(a) (b) (c) (d)

Fig. 1. Compare the performances of Algorithm $\mathcal{A}$ and $\mathcal{B}$. Images (a) and (b) are the original 8 bpp image and the hand-labeled classified image. Image (c): Algorithm $\mathcal{A}$ with classification error rate 15.8%. Image (d): Algorithm $\mathcal{B}$ with classification error rate 15.3%.

Estimation algorithm	Sensitivity	Specificity	PVP	$\mathcal{P}_e$
$\mathcal{A}$ (N=8)	0.8034	0.7509	0.7728	0.2168
$\mathcal{A}$ (N=16)	0.8044	0.7615	0.7769	0.2149
$\mathcal{A}$ (N=32)	0.8012	0.7856	0.8013	0.1981
$\mathcal{B}$	0.7151	0.8774	0.8625	0.2011

Table 1. The table compares the performance of algorithms $\mathcal{A}$ and $\mathcal{B}$ for 2-D image classification.

Estimation algorithm	Training time	Classification time
$\mathcal{A}$ (N=8)	36s	1s
$\mathcal{A}$ (N=16)	127s	5s
$\mathcal{A}$ (N=32)	474s	19s
$\mathcal{B}$	4s	1s

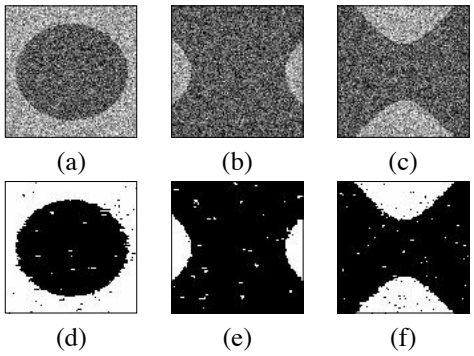
Table 2. The table compares the running times of algorithms $\mathcal{A}$ and $\mathcal{B}$ for 2-D image classification.

denoted by $\{\mathcal{D}_{i,j} : i, j \in (0, 1, 2, 3)\}$, the definitions of the 6 features used are:

1. $f_1 = \mathcal{D}_{0,0}$, $f_2 = |\mathcal{D}_{1,0}|$, $f_3 = |\mathcal{D}_{0,1}|$,
2. $f_4 = \sum_{i=2}^{3} \sum_{j=0}^{1} |D_{i,j}|/4$,
3. $f_5 = \sum_{i=0}^{1} \sum_{j=2}^{3} |D_{i,j}|/4$,
4. $f_6 = \sum_{i=2}^{3} \sum_{j=2}^{3} |D_{i,j}|/4$,

Figure 1 compares the classification performances of algorithms $\mathcal{A}$ and $\mathcal{B}$ for one particular image from the dataset. The original and hand labeled images are also shown and the error rates are indicated in the figure. A six-fold cross validation was used to compare the classification performances of the algorithms. At each iteration, one image was used as test data and the other five were used as training data. The man-made class was assumed to be the target class, or the positive class for calculating *sensitivity*, *specificity*, and *predictive positive value (PVP)*. Table 1 shows the average performances of algorithms $\mathcal{A}$ and $\mathcal{B}$ over all 6 iterations. All the experiments were performed on 2.6 GHz Xeon processors running Linux. The CPU times required by Algorithms $\mathcal{A}$ and $\mathcal{B}$ for learning and classification respectively are shown in Table 2.

We notice from Tables 1 and 2 that as N is increased for Algorithm $\mathcal{A}$, the error rate decreases but the model learning and classification times go up. The average error rates ($\mathcal{P}_e$) using algorithms $\mathcal{A}$ and $\mathcal{B}$ are found to be comparable. However, the running times for learning as well as classification, using Algorithm $\mathcal{B}$, are much less compared to Algorithm $\mathcal{A}$.



(a) (b) (c) (d) (e) (f)

Fig. 2. Segmentation of a hyperboloid with 3-D HMM. The figure shows 3 frames one each in the $X-Y$, $Y-Z$ and $X-Z$ planes. (a)-(c) is the original image; (d)-(f) is the segmented image. The parameters $a = 35$, $b = 25$, $c = 30$, and $\sigma = 0.6$. Error rate is 1.7%.

5. UNSUPERVISED SEGMENTATION OF 3-D IMAGES USING 3-D HMM

The 3-D HMM has been applied to volume image segmentation. Experiments were performed using a large pool of synthetic volume images. Each image contained a standard mathematical 3-D shape, centered about the image. In particular we used the shapes *torii*, *ellipsoids*, and *hyperboloids*, the mathematical equations of which were obtained from *Mathworld* [1].

The method of image generation was as follows. Points in the interior of the 3-D shape were assigned black color while the rest were white. Each color voxel, black ($\mu = 0$) and white ($\mu = 1$), was perturbed by an additive Gaussian

[1] `http://mathworld.wolfram.com`

parameters		α					
a	c	0.0	0.2	0.4	0.6	0.8	1.0
20	20	0.0144	0.0116	0.0068	0.0072	**0.0064**	0.0816
20	30	**0.0104**	0.0114	0.0110	0.0106	0.0105	0.0276
20	40	0.0129	0.0124	0.0120	**0.0115**	0.0119	0.0372
30	30	0.0365	0.0342	**0.0314**	0.0338	0.0372	0.0373
30	40	0.0338	0.0598	0.0449	0.0511	**0.0233**	0.0353
40	40	**0.0091**	0.0115	0.0468	0.0184	0.0687	0.0456

Table 3. The table compares the segmentation performances of a few images of class torii with shown shape parameters (size $100\times100\times100$ and $\sigma = 0.5$) as the model parameter α is varied between 0 and 1. The best performance is indicated in bold.

noise $\sim N(0, \sigma^2)$ and the voxel values were truncated to lie in the interval $[-2\sigma, 1 + 2\sigma]$. For the purpose of displaying images, voxel values in the interval $[-2\sigma, 1 + 2\sigma]$ were scaled to $[0, 255]$. A unidimensional feature was used for each color voxel. The 3-D shape parameters (length of semi axes and radii, denoted by a, b, and c) and the noise parameter σ were varied to form a pool of 70 images and studies were performed to determine the sensitivity of the 3-D HMM algorithm to the various parameters. They are explained as follows:

1. In [6], a model regularization was proposed to regularize 3-D transition probabilities towards 2-D transition probabilities. A parameter α determined the extent of 3-D dependence. Here, α was varied between 0 and 1 in steps of 0.2 and segmentation performance noted for each α. The number of states were fixed at 3 for each class. Results for $\sigma = 0.5$ for torii are shown in Figure 3. The best performance, which usually occurs at an intermediate value of α, is indicated in bold. A trade-off between model complexity (complete 3-D model, $\alpha = 1$) and ease of estimation (2-D model, $\alpha = 0$), is preferred in most cases, and the results support this hypothesis. Due to space limitation, we have shown few results here. More complete results will be soon available online [2] and in our journal paper.

2. The Gaussian noise parameter σ was varied between 0.2 and 0.7 in steps of 0.1. The number of states were fixed at 3 for each class and α was fixed at 0.6. Segmentation was performed for all 70 images, for each σ. The best and median segmentation performances, for different values of σ, are shown in Table 4. As is evident from the results, 3-D HMM performs reasonably well segmentation even for large values of σ.

The running times of 3-D HMM segmentation program for image sizes ($w \times w \times w$), where w takes values 50, 100, 150, and 200, were found out to be 32s, 280s, 798s, and 938s respectively. These numbers support the fact that the complexity of the algorithm is linear in problem size (w^3).

[2]URL: `http://wang.ist.psu.edu/3DHMM`

σ	0.2	0.3	0.4	0.5	0.6	0.7
$\mathcal{P}_{best}$	0.0001	0.0005	0.0019	0.0037	0.0067	0.0058
$\mathcal{P}_{med}$	0.0004	0.0041	0.0157	0.0406	0.1207	0.1995

Table 4. The table compares the best ($\mathcal{P}_{best}$) and median ($\mathcal{P}_{med}$) segmentation performances over 70 images (size $100\times100\times100$) as the variance of the Gaussian noise varies.

6. CONCLUSIONS

In this paper, we implemented a computationally fast parameter estimation technique for 2-D HMMs and discussed its scalability for multi-dimensional HMMs. Supervised classification of 2-D aerial images was performed and comparisons with the original estimation algorithm showed that the proposed algorithm is much faster. A parametric study was performed with 3-D HMM. In the future, we would like to incorporate the proposed algorithm into 2-D multiresolution HMM based applications. Multiresolution extension of 3-D HMMs is also of interest to us.

7. REFERENCES

[1] S. Eickeler, S. Muller, G. Rigoll, "Improved face recognition using pseudo 2-D hidden Markov models," *Proc. Workshop on Advances in Facial Image Analysis and Recognition Technology in conjunction with ECCV*, Freiburg, Germany, June 1998.

[2] F. Hulsken, F. Wallhoff, G. Rigoll, "Facial expression recognition with pseudo-3D hidden Markov models," *Proc. DAGM-Symposium, LNCS,* vol. 2191, pp. 291-297, 2001.

[3] J. Li, A. Najmi, R. M. Gray, "Image classification by a two dimensional hidden Markov model," *IEEE Trans. Signal Processing,* vol. 48, no. 2, pp. 517-33, February 2000.

[4] J. Li, R. M. Gray, R. A. Olshen, "Multiresolution image classification by hierarchical modeling with two dimensional hidden Markov models," *IEEE Trans. Information Theory,* vol. 46, no. 5, pp. 1826-41, August 2000.

[5] J. Li, J. Z. Wang, "Automatic linguistic indexing of pictures by a statistical modeling approach, " *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 25, no. 9, pp. 1075-1088, 2003.

[6] J. Li, D. Joshi, and J. Z. Wang, "Stochastic modeling of volume images with a 3-D hidden Markov model," *Proc. IEEE Int. Conf. on Image Processing (ICIP)*,pp. 2359-2362, Singapore, October 2004.