IMAGE CLASSIFICATION BY A TWO DIMENSIONAL HIDDEN MARKOV MODEL

Jia Li, Amir Najmi and Robert M. Gray

Information Systems Laboratory, EE Dept.
Stanford University, CA 94305
jiali@isl.stanford.edu,
zaalim@leland.stanford.edu
gray@ee.stanford.edu

ABSTRACT

Traditional block-based image classification algorithms, such as CART and VQ based classification, ignore the statistical dependency among image blocks. Consequently, these algorithms often suffer from over-localization. In order to benefit from the inter-block dependency, an image classification algorithm based on a hidden Markov model (HMM) is developed. An HMM for image classification, a two dimensional extension from the one dimensional HMM used for speech recognition, has transition probabilities conditioned on the states of neighboring blocks from both directions. Thus, the dependency in two dimensions can be reflected simultaneously. The HMM parameters are estimated by the EM algorithm. A two dimensional version of the Viterbi algorithm is also developed to classify optimally an image based on the trained HMM. An application of the HMM algorithm to document image and aerial image segmentation shows that the algorithm performs better than CART.

1. INTRODUCTION

For most block based image classification algorithms, such as CART [1], images are divided into blocks and decisions are made independently for the class of each block. This approach leads to an issue of choosing block sizes. We do not want to choose a too large block size since this obviously causes crude classification. On the other hand, if we choose a small block size, only very local properties belonging to the small block are examined in classification. The penalty then comes from losing information about surrounding regions. A well known method in signal processing to attack this type of problem is to use context information. Trellis coding [2] in image compression is such an example. How to introduce "the incorporation of context" into classifiers is what is of interest to us. Previous work [3] has looked into ways of taking advantage of context information to improve classification performance for document image segmentation. Both block sizes and classification rules can vary according to context. The great improvement achieved demonstrates the potential of context information to help classification. The purpose of this paper is to introduce a two dimensional hidden Markov model (2-D HMM) as a general framework to build context dependent classifiers.

Hidden Markov models have earned their popularity mostly from successful application to speech recognition [4, 5, 6]. Despite the weakness of the Markovian assumption as applied to speech, they have proven to be a powerful method in speech processing. The probability mechanism is as follow: at any discrete unit of time, the system is assumed to exist in one of a finite set of states. Within each state there is a fixed probability distribution of generating a single observation (feature vector). This probability distribution is typically modeled as a mixture of Gaussian distributions. Transitions between states take place according to a fixed probability depending only on the state of the system at the unit of time immediately preceding (1-step Markovian). HMMs owe both their name and modeling power to the fact that these states represent abstract quantities and are themselves never observed. They correspond to "clusters" of contexts having similar probability distributions of the observed feature vector. Thus the number of states is a meta-parameter to be chosen in the design.

In this paper, we extend the idea of the HMM to image classification. Since image data is two-dimensional, the probability of the system entering a particular state depends upon the state of the system at the adjacent observations in both horizontal and vertical directions. As in the case of speech, the probability distribution of the feature vector is modeled as a fixed Gaussian distribution for any given state. The main difficulty lies in finding efficient methods to build and apply a two-dimensional model. Several techniques are explored here that are required to make the two-dimensional extension computationally feasible.

In Section 2, we provide a mathematical formulation of the basic assumptions of HMM. The algorithm is presented in Section 3. Section 4 discusses techniques to speed up the algorithm, so that it is computationally feasible. The results are given in Section 5. We conclude in Section 6.

This work was supported by the National Science Foundation under NSF Grant No. MIP-931190 and by gifts from Hewlett-Packard, Inc., and SK Telecom, Inc.

2. BASIC ASSUMPTIONS OF 2-D HMM

As in all block based classification systems, an image to be classified is divided into blocks and feature vectors are evaluated as the statistics of the blocks. The image is then classified according to the feature vectors.

The 2-D HMM assumes that the feature vectors are generated by a Markov model which may change state once every block. Suppose there are M states, the state of block (i,j) is denoted by $s_{i,j}$. The feature vector of block (i,j) is $\mathbf{x}_{i,j}$ and the class is $c_{i,j}$. We use $P(\cdot)$ to represent the probability of an event. We denote $(i',j') < (i,j)$ if $i' < i$ or $i' = i, j' < j$; in which case we say that block (i',j') is before block (i,j). For example, the blocks before (i,j) in Fig. 1 are the shaded blocks. This sense of order is the same as the raster order of row by row. We would like to point out, however, that we introduce this order only for stating the assumptions. In classification, we do not classify blocks one by one in such an order. Our classification algorithm tries to find the optimal combination of classes for many blocks jointly. A one dimensional approach of joint classification, assuming a scanning order in classification, is usually suboptimal.

The first assumption we make is that

$$\begin{aligned} & P(s_{i,j}|\text{context}) = a_{m,n,l} \quad , \\ \text{where} \quad & \text{context} = \{s_{i',j'}, \mathbf{x}_{i',j'}, (i',j') < (i,j)\} \\ \text{and} \quad & m = s_{i-1,j},\ n = s_{i,j-1},\ \text{and}\ l = s_{i,j}\ . \end{aligned}$$

The above assumption can be summarized by the following two points. First, the state $s_{i',j'}$ is a sufficient statistic of $(s_{i',j'}, \mathbf{x}_{i',j'})$ for estimating transition probabilities. Second, the state transition is first order Markovian in a two dimensional sense. Shown in Fig. 1, knowing the states of all the shaded blocks, we only need the states of the two adjacent blocks in the darker shade to calculate the transition probability to a next state. We also assume that there is a unique mapping from states to classes. Thus, the classes of the blocks are determined once the states are known.

The second assumption is that for every state, the feature vectors follow a Gaussian mixture distribution. Once the state of a block is known, the feature vector is conditionally independent of the other blocks. Since any state with an M-component Gaussian mixture can be split into M substates with single Gaussian distributions, we constrain to single Gaussian distributions in our model. For a block with state s and feature vector $\mathbf{x}$, the distribution is

$$b_s(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^n |\mathbf{\Sigma}_s|}} e^{-\frac{1}{2}(\mathbf{x}-\mu_s)' \mathbf{\Sigma}_s^{-1} (\mathbf{x}-\mu_s)} \ ,$$

where $\mathbf{\Sigma}_s$ is the covariance matrix and μ_s is the mean vector.

The task of our classifier is to estimate the 2-D HMM from training data and to classify images by finding the combination of states with the maximum posterior probability given the observed feature vectors.

3. THE ALGORITHM

For the assumed HMM, we need to estimate the following parameters: transition probabilities $a_{m,n,l}$, where $m,n,l = 1,...,M$ and M is the total number of states, the mean μ_m, and the covariance matrix $\mathbf{\Sigma}_m$ of the Gaussian distributions, $m = 1,...,M$. The parameters are estimated by the EM algorithm [7, 8, 9]. Specific to our model, the algorithm iteratively improves the model estimation by the following two steps.

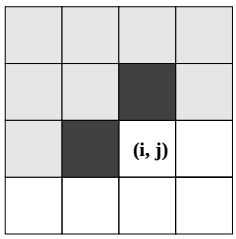

Figure 1: Markovian property of the transition of states

1. Given the current model estimation $\phi^{(p)}$ and the observed feature vectors $\mathbf{x}_{i,j}$, the mean vectors and covariance matrices are updated by

$$\begin{aligned} \mu_m^{(p+1)} &= \frac{\Sigma_{i,j} L_m^{(p)}(i,j) \mathbf{x}_{i,j}}{\Sigma_{i,j} L_m^{(p)}(i,j)} \\ \mathbf{\Sigma}_m^{(p+1)} &= \frac{\Sigma_{i,j} L_m^{(p)}(i,j) (\mathbf{x}_{i,j} - \mu_m^{(p)})(\mathbf{x}_{i,j} - \mu_m^{(p)})'}{\Sigma_{i,j} L_m^{(p)}(i,j)} \ , \end{aligned}$$

   where $L_m^{(p)}(i,j)$ is the likelihood of being in state m at block (i,j) given the observed feature vectors, classes and model $\phi^{(p)}$.

2. The transition probabilities are updated by

$$a_{m,n,l}^{(p+1)} = \frac{\Sigma_{i,j} H_{m,n,l}^{(p)}(i,j)}{\Sigma_{i,j} L_l^{(p)}(i,j)} \ ,$$

   where $H_{m,n,l}^{(p)}(i,j)$ is the likelihood of being in state m at block $(i-1,j)$, state n at block $(i,j-1)$, and state l at block (i,j) given the observed feature vectors, classes, and model $\phi^{(p)}$.

In the case of one dimensional HMM as used in speech recognition, computationally efficient formulas exist for calculating $L_m(k)$ and $H_{m,l}(k)$ [6]. For 2-D HMM, however, the computation of $L_m(i,j)$ and $H_{m,n,l}(i,j)$ is not feasible, due to the two dimensional transition probabilities. The next section will discuss why this is so and how to reduce the computational complexity.

4. COMPUTATIONAL COMPLEXITY

The EM procedure outlined in the previous section is designed to choose HMM parameters μ_m, Σ_m and $a_{m,n,l}$ that maximize the likelihood of the observed features (training data) unconditional on the sequence of states $s_{i,j}, (i,j) \in \mathbb{N}$ taken by the system, where $\mathbb{N} = \{(i,j), 0 \le i < \mathfrak{m}, 0 \le j < \mathfrak{n}\}$ denotes the collection of all the blocks in an image. This is just a sum of the likelihoods of the observation conditioned on each state sequence weighted by the probability of each state sequence. However, to simplify the calculation

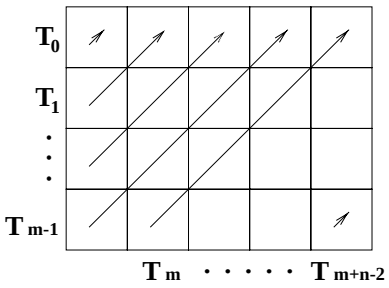

Figure 2: Blocks on diagonals of an image

of $L_m(i,j)$ and $H_{m,n,l}(i,j)$, we will assume that the single most likely state sequence accounts for virtually all the likelihood of the observations. Thus we find the optimal state sequence which maximizes $P(s_{i,j}, \mathbf{x}_{i,j}, c_{i,j},\ (i,j) \in \mathbb{N})$. This is what is called Viterbi training in [6]. The calculation of this probability is almost the same as $P(s_{i,j}, \mathbf{x}_{i,j},\ (i,j) \in \mathbb{N})$. If the classes corresponding to $s_{i,j}$, $(i,j) \in \mathbb{N}$, are the same as $c_{i,j}$, the two probabilities are equal. Otherwise, $P(s_{i,j}, \mathbf{x}_{i,j}, c_{i,j},\ (i,j) \in \mathbb{N}) = 0$. Hence, we only discuss the computation of $P(s_{i,j}, \mathbf{x}_{i,j},\ (i,j) \in \mathbb{N})$ in detail here. When we apply the trained model to classify images, the maximization of $P(s_{i,j}, \mathbf{x}_{i,j},\ (i,j) \in \mathbb{N})$ is equivalent to maximizing the posterior probability of the combination of states given the feature vectors.

According to the 2-D HMM assumptions we made, we can calculate $P(s_{i,j}, \mathbf{x}_{i,j},\ (i,j) \in \mathbb{N})$ by an efficient formula as below.

$$\begin{aligned} P(s_{i,j}, \mathbf{x}_{i,j},\ (i,j) \in \mathbb{N}) &= P(s_{i,j},\ (i,j) \in \mathbb{N}) \cdot \\ &\quad P(\mathbf{x}_{i,j},\ (i,j) \in \mathbb{N} \,|\, s_{i,j},\ (i,j) \in \mathbb{N}) \\ &= P(s_{i,j},\ (i,j) \in \mathbb{N}) \cdot \prod_{(i,j)\in\mathbb{N}} P(\mathbf{x}_{i,j} \,|\, s_{i,j}) \end{aligned}$$

The probability of a state sequence of the image can be calculated by:

$$\begin{aligned} P(s_{i,j}, (i,j) \in \mathbb{N}) &= P(T_0) \cdot P(T_1|T_0) \cdot \\ &\quad P(T_2|T_1) \cdots P(T_{\mathrm{m+n-2}}|T_{\mathrm{m+n-3}}), \end{aligned}$$

where T_i denotes the sequence of states for blocks lying on diagonal i, i.e., $(s_{i,0}, s_{i-1,1}, \cdots, s_{0,i})$, as shown in Fig. 2.

We can see that T_i serves as an "isolating" element in the expansion of $P(s_{i,j}, (i,j) \in \mathbb{N})$ because of the 2-D Markovian property of our model. Thus the Viterbi algorithm can be straightforwardly applied to find the combination of states which maximizes the probability. The difference from the normal Viterbi algorithm is that the number of possible sequences of states at every position in the Viterbi transition diagram increases exponentially with the increase of blocks in T_i. If there are M states, the amount of computation and memory are both in the order of M^k, where k is the number of blocks in T_i. Fig. 3 shows an example. Hence, we refer to this version of the Viterbi algorithm as the 2-D Viterbi algorithm.

To reduce computation, at every position of the Viterbi transition diagram, the algorithm only uses N out of all the M^k sequences of states, shown in Fig. 4. The paths are

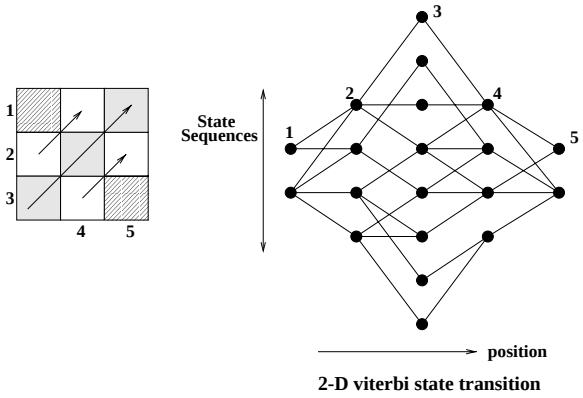

Figure 3: The 2-D Viterbi algorithm

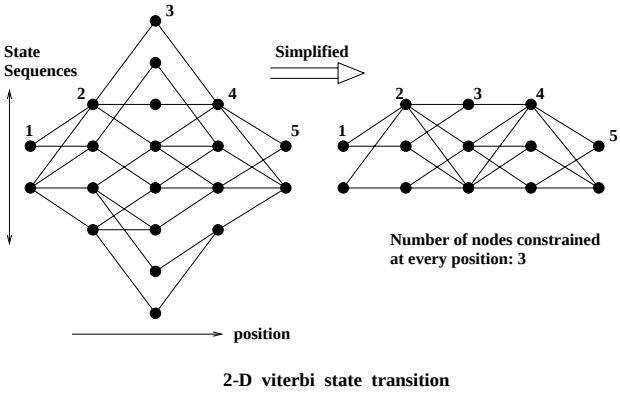

Figure 4: The computation reduced Viterbi algorithm

constrained to pass one of these N nodes. To choose the N sequences of states, the algorithm separates the blocks in the diagonal from the other blocks by ignoring their statistical dependency. Consequently, the posterior probability of a sequence of states on the diagonal is evaluated as a product of the posterior probability of every block. Then, the N sequences with the largest posterior probabilities are chosen as the N nodes allowed in the Viterbi transition diagram. The implicit assumption for doing this is that the optimal state sequence (the node in the optimal path of the Viterbi transition diagram) yields high likelihood when the blocks are treated independently. We also expect that when the optimal state sequence is not among the N nodes, the chosen suboptimal state sequence coincides with the optimal sequence at most of the blocks. A fast algorithm is developed for choosing such N sequences of states. We do not need to calculate the posterior probabilities of all the M^k sequences in order to choose the largest N from them.

After the model is trained, to classify an image, we use the fast version of the 2-D Viterbi algorithm to find the collection of states with nearly maximum posterior probability given the feature vectors. Once the states are determined, the classes of the image can be obtained by a simple mapping from the states.

5. RESULTS

The first application of our algorithm is the segmentation of man-made and natural regions of aerial images. We divide the images into 4×4 blocks and use the DCT coefficients or the averages over some of them as features. We compare the 2-D HMM result with that obtained by CART [1]. The basic idea of CART is to partition a feature space by a tree structure and assign a class to every cell of the partition. Feature vectors landing in a cell are classified as the class of the cell. Using CART, we obtain error rate of 21.12%. However, the 2-D HMM algorithm achieves error rate of 14.68%.

We also applied our algorithm to the text and photograph segmentation of document images. By pictures, we mean continuous-tone images such as photographs. By text, we mean normal text, tables and graphs [10]. The features we use are described in detail in [10]. The original image and the manually classified image are in the upper panel of Fig. 5. The classification results of both CART and the 2-D HMM algorithm are shown in the lower panel of Fig. 5. We can see that the result using HMM is much cleaner than the result using CART, especially in the picture regions. This is expected since the classification based on HMM takes context into consideration. As a result, some smooth blocks in the picture regions, which locally resemble text blocks can be correctly identified as picture.

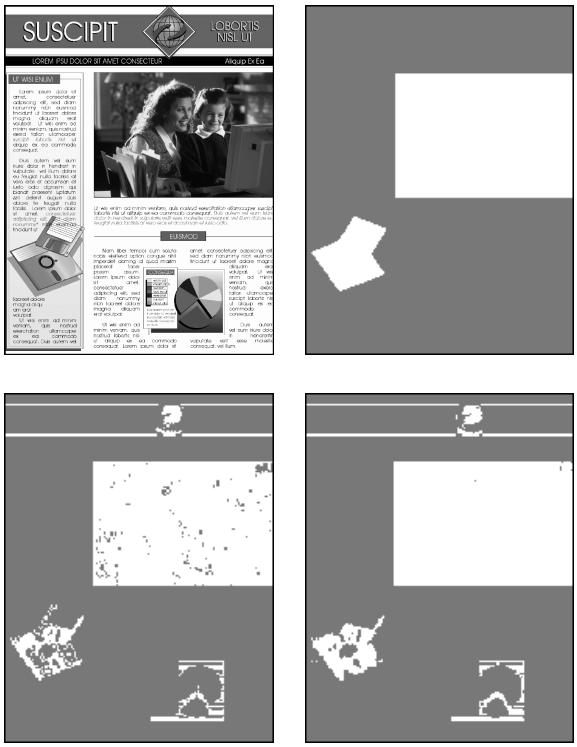

Figure 5: Comparison of the classification results of CART and 2-D HMM. Upper: an image and its hand labeled classes (gold standard). Lower left: CART classification result. Lower right: 2-D HMM classification result. White: photograph, Gray: text.

For the 2-D HMM algorithm, in both applications, we assign 4 states to each of the two classes. Simulation shows that models with around 4 states per class give very similar results.

6. CONCLUSIONS

We propose a two dimensional hidden Markov model for image classification. The two dimensional model provides a structured way to use context information in classification. As the model is two dimensional, computational complexity is an important issue. We describe fast algorithms to efficiently estimate the model and to perform classification based on the model. The application of the algorithm to several problems shows better performance than that of existing algorithms.

7. REFERENCES

[1] L. Breiman, J. H. Friedman, R. A. Olshen and C. J. Stone, *Classification and Regression Trees,* Chapman & Hall, 1984.

[2] Allen Gersho and Robert M. Gray, *Vector Quantization and Signal Compression*, pages 555-585, Kluwer Academic Publishers, 1992.

[3] Jia Li and Robert M. Gray, "Context Based Multiscale Classification of Images," *Proceedings of International Conference on Image Processing*, Chicago, Oct. 1998.

[4] X. D. Huang, Y. Ariki and M. A. Jack, *Hidden Markov models for speech recognition*, Edinburgh University Press, 1990.

[5] R. Cole, L. Hirschman, L. Atlas, et al., "The challenge of spoken language systems: research directions for the nineties," *IEEE Transactions on Speech and Audio Processing*, volume 3, pages 1-21, 1063-6676, Jan. 1995.

[6] S. Young, J. Jansen, J. Odell, D. Ollason and P. Woodland, *HTK - Hidden Markov Model Toolkit*, Cambridge University, 1995.

[7] A. P. Dempster, N. M. Laird and D. B. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm," *J. Roy. Statist. Soc.*, 1997.

[8] C. F. J. Wu, "On the Convergence Properties of the EM Algorithm," *The Annals. of Statistics*, 1983.

[9] L. E. Baum, T. Petrie, G. Soules, and N. Weiss, "A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains," *The Annals of Math. Stat.*, 1970.

[10] Jia Li and Robert M. Gray, "Text and Picture Segmentation by the Distribution Analysis of Wavelet Coefficients," *Proceedings of International Conference on Image Processing*, Chicago, Oct. 1998.