EURASIP Journal on Applied Signal Processing 2005:13, 1984–1993

Fast Schemes for Computing Similarities between Gaussian HMMs and Their Applications in Texture Image Classification

Ling Chen

Department of Electrical and Computer Engineering, Stevens Institute of Technology, Castle Point on Hudson, Hoboken, NJ 07030, USA
Email: lchen@stevens-tech.edu

Hong Man

Department of Electrical and Computer Engineering, Stevens Institute of Technology, Castle Point on Hudson, Hoboken, NJ 07030, USA
Email: hman@stevens-tech.edu

Received 31 December 2003; Revised 18 August 2004

An appropriate definition and efficient computation of similarity (or distance) measures between two stochastic models are of theoretical and practical interest. In this work, a similarity measure, that is, a modified "generalized probability product kernel," of Gaussian hidden Markov models is introduced. Two efficient schemes for computing this similarity measure are presented. The first scheme adopts a forward procedure analogous to the approach commonly used in probability evaluation of observation sequences on HMMs. The second scheme is based on the specially defined similarity transition matrix of two Gaussian hidden Markov models. Two scaling procedures are also proposed to solve the out-of-precision problem in the implementation. The effectiveness of the proposed methods has been evaluated on simulated observations with predefined model parameters, and on natural texture images. Promising experimental results have been observed.

Keywords and phrases: similarity measure, hidden Markov model, kernel method, Bhattacharyya affinity, texture classification.

1. INTRODUCTION

Hidden Markov model (HMM) has been adopted in a wide variety of application areas including econometrics, computational biology, statistical process control, and speech recognition. Recently, it was also introduced to image processing applications such as face recognition [1, 2] and texture analysis [3, 4]. A challenging problem for HMM is that given their model parameters, how to define an appropriate similarity (or distance) measure for two HMMs [5].

An appropriate similarity measure between two HMMs is of theoretical interests and it can also be a useful tool in various applications. For instance, in computational biology, because of the availability of large libraries of profile HMMs, there exists the possibility of comparing sequence families by comparing the profiles of the families rather than comparing the individual members of the families. It is also possible to compare a sequence family instead of its individual members towards an HMM that is trained to model a particular feature [6]. In image retrieval applications, each texture image can be modeled by a wavelet-domain HMM and the classification is carried out by computing distances between the model of the query image and those of all candidate images [4].

There have been some research efforts on this problem and several techniques are proposed in the literature [4, 5, 6, 7, 8]. To begin with, denote $\lambda = (A, B, \pi)$ as the model parameters of an HMM, where A is a *state transition distribution*, B is the *observation probability distribution*, and π is the *initial state distribution*. Denote $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T)$ as an observation sequence generated by λ. Observations $\mathbf{x}$ can be either discrete symbols chosen from a finite alphabet ($\mathbf{x} \in \mathbf{V} = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_M\}$) or continuous (vector) signals ($\mathbf{x} \in \mathbb{N}^D$). In [5], a distance measure between two HMMs, λ and λ', was proposed as

$$D(\lambda, \lambda') = \frac{1}{T}\left[\log P(\mathbf{X}'|\lambda) - \log P(\mathbf{X}'|\lambda')\right], \tag{1}$$

where $\mathbf{X}'$ is an observation sequence generated by the model λ'. A symmetrized version of this distance measure is

$$D_s(\lambda, \lambda') = \frac{D(\lambda, \lambda') + D(\lambda', \lambda)}{2}. \tag{2}$$

In [6], the *coemission probability* of two profile HMMs is defined as

$$\sum_{\mathbf{X}\in\mathbf{V}\times\cdots\times\mathbf{V}} P(\mathbf{X}|\lambda)P(\mathbf{X}|\lambda'), \tag{3}$$

where λ and λ' are two profile HMMs. In [4], the distance between the two HMMs is computed based on Kullback-Leibler (KL) distance:

$$D(\lambda,\lambda') = \int_{\mathbb{N}^D\times\cdots\times\mathbb{N}^D} P(\mathbf{X}|\lambda)\log\frac{P(\mathbf{X}|\lambda)}{P(\mathbf{X}|\lambda')}d\mathbf{X}. \tag{4}$$

Recently, a *generalized probability product kernel* (GPPK) between distributions, which represents the similarity between two probability distributions p and p', is proposed [8]:

$$K_\rho(p,p') = \int_\Omega p(x)^\rho p'(x)^\rho dx, \tag{5}$$

where normally $\rho \in \{1/2, 1, 2, 3, \ldots\}$. The motivation behind this method is to combine discriminative and generative estimations to exploit their complementary advantages. In [8], the *expected likelihood kernel*, that is, $K_1(p,p')$, is used to derive the similarity measure between Gaussian mixture models, HMMs, and so forth. Some of the advantages of the GPPK include the positive-definite, symmetric property and the capability to handle a variety of generative models (including HMMs) in a closed form (note that KL distance is not positive definite, asymmetric and in many cases, it can only be approximated by an upper bound). However, the closed-form evaluation of GPPK is not readily available except for $\rho = 1$. Whereas other values of ρ, especially $\rho = 1/2$, can also be of great interest in some cases, notice that when $\rho = 1/2$, the GPPK is effectively the well-known Bhattacharyya's measure of affinity between two statistical distributions [9]. In this work, we develop a modified GPPK to efficiently evaluate similarity between HMMs in closed form whereas the value of ρ can be chosen freely, that is, not constrained to 1. The modification is developed based on a new interpretation of GPPK. Under this new interpretation, the similarity measure of two HMMs is considered as the statistical average of similarities of all possible so-called *costate sequences* drawn from the two HMMs.

Meanwhile, the brute-force computation of the similarity between two Gaussian HMMs can be prohibitively intensive, for example, the computational complexity is $O(3T(NN')^{T+1})$, where T is the number of transitions and N and N' are the numbers of states of two HMMs. In this work, we propose two fast schemes which can drastically ease this burden by reducing the computational complexity to $O(3T(NN')^2)$ and $O((NN')^3\log_2 T)$, respectively. The relative computational complexity of these two schemes depends on the complexities of the two HMMs, for example, the number of states N and the number of transitions T to be used in the evaluation process. It can be measured by the ratio of $3T/NN'\log_2 T$.

Another important implementation issue of the proposed similarity measure is the out-of-precision problem. Because the computation of similarity between HMMs will exceed the precision limit of any machines when T gets large, we formulate two scaling procedures corresponding to two proposed fast schemes. The scaling procedures can prevent the computation from going beyond the precision range as well as guarantee that the exact value of the similarity measure can be evaluated.

The paper is arranged as follows. Section 2 introduces the similarity measure of Gaussian HMMs and its modification. Section 3 presents the forward procedure and its scaling procedure for computing the proposed similarity measure. Section 4 presents the second fast scheme based on similarity transition matrix, which is followed by the scaling procedure. Section 5 provides the preliminary experimental results. We conclude our work in Section 6.

2. SIMILARITY MEASURE OF GAUSSIAN HMMS AND ITS MODIFICATION

One of the building blocks in deriving the similarity measure of general HMMs is computing the similarity measure of any arbitrary pair of observation distributions corresponding to two specific states of these two HMMs, denoted as $\psi_{s,s'}$. For Gaussian HMMs, the observation probability distribution is Gaussian. Based on (5), the GPPK of two D-dimensional Gaussians, $p(\mathbf{x}) \sim \mathcal{N}(\boldsymbol{\mu},\boldsymbol{\Sigma})$ and $p'(\mathbf{x}) \sim \mathcal{N}'(\boldsymbol{\mu}',\boldsymbol{\Sigma}')$, is computed as

$$\begin{aligned}\psi_{\mathcal{N},\mathcal{N}'} &\triangleq K_\rho(p,p') \\ &= \int_{\mathbb{R}^D} p(\mathbf{x})^\rho p'(\mathbf{x})^\rho d\mathbf{x} \\ &= (2\pi)^{(1-2\rho)D/2}|\boldsymbol{\Sigma}^\dagger|^{1/2}|\boldsymbol{\Sigma}|^{-\rho/2}|\boldsymbol{\Sigma}'|^{-\rho/2} \\ &\quad\times\exp\left(-\frac{\rho}{2}\boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}-\frac{\rho}{2}\boldsymbol{\mu}'^{\mathrm{T}}\boldsymbol{\Sigma}'^{-1}\boldsymbol{\mu}'+\frac{1}{2}\boldsymbol{\mu}^{\dagger\mathrm{T}}\boldsymbol{\Sigma}^\dagger\boldsymbol{\mu}^\dagger\right),\end{aligned} \tag{6}$$

where $\boldsymbol{\Sigma}^\dagger = (\rho\boldsymbol{\Sigma}^{-1}+\rho\boldsymbol{\Sigma}'^{-1})^{-1}$ and $\boldsymbol{\mu}^\dagger = \rho\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}+\rho\boldsymbol{\Sigma}'^{-1}\boldsymbol{\mu}'$. Note that the computational complexity of $\psi_{\mathcal{N},\mathcal{N}'}$ is mainly determined by the complexity of matrix determinants and inverses (which both are $O(D^3)$) in (6).

For Gaussian HMM, given the observation sequence $\mathbf{X}$ and the model parameters λ, the likelihood is

$$P(\mathbf{X}|\lambda) = \sum_{s_0,\ldots,s_T=1}^{N}\pi_{s_0}b(\mathbf{x}_0|s_0)\prod_{t=1}^{T}b(\mathbf{x}_t|s_t)a_{s_t|s_{t-1}}, \tag{7}$$

where π_{s_0} is the initial state probability of state s_0, $b(\mathbf{x}_t|s_t)$ is the Gaussian distribution corresponding to state s_t, and $a_{s_t|s_{t-1}}$ is the state transition probability from state s_{t-1} to s_t. When $\rho = 1$, the GPPK of two Gaussian HMMs is

$$\begin{aligned}&K_\rho(\lambda,\lambda') \\ &\quad= \int_{\mathbb{R}^D\times\cdots\times\mathbb{R}^D} P(\mathbf{X}|\lambda)P(\mathbf{X}|\lambda')d\mathbf{X} \\ &\quad= \sum_{s_0,\ldots,s_T=1}^{N}\sum_{s'_0,\ldots,s'_T=1}^{N'}\pi_{s_0}\pi_{s'_0}\psi_{s_0,s'_0}\prod_{t=1}^{T}a_{s_t|s_{t-1}}a_{s'_t|s'_{t-1}}\psi_{s_t,s'_t},\end{aligned} \tag{8}$$

where ψ_{s_t,s'_t}, $t = 0, 1, 2, \dots, T$, is the GPPK of two Gaussians corresponding to states s_t and s'_t of two HMMs.[1]

The derivation of (8) is based on the definition of GPPK with the value of ρ being set to 1. Here we take another perspective to derive the similarity measure that is exactly the same as the one depicted in (8), whereas the value of ρ is no longer constrained to be 1. For two HMMs λ and λ', suppose from $t = 0$ to $t = T$ that the state sequence is $\mathbf{s} = (s_0, s_1, s_2, \dots, s_T)$ and $\mathbf{s}' = (s'_0, s'_1, s'_2, \dots, s'_T)$. Then we define a "costate" sequence as $\mathbf{ss}' = (s_0, s'_0, s_1, s'_1, \dots, s_T, s'_T)$. The probability of such a costate sequence $\mathbf{ss}'$ with model parameters λ and λ' can be computed as

$$P(\mathbf{ss}'|\lambda,\lambda') = P(\mathbf{s}|\lambda)P(\mathbf{s}'|\lambda') = \pi_{s_0}\pi_{s'_0}\prod_{t=1}^{T} a_{s_t|s_{t-1}}a_{s'_t|s'_{t-1}}. \tag{9}$$

Define the similarity of two state sequences as

$$\psi_{\mathbf{ss}'} = \prod_{t=0}^{T}\psi_{s_t,s'_t}. \tag{10}$$

The value of $\psi_{\mathbf{ss}'}$ is determined by the value of ρ used in the computation of ψ_{s_t,s'_t}, that is, the GPPK of two Gaussians corresponding to two states s_t and s'_t of two Gaussian HMMs (see (6)). Then the similarity measure between λ and λ', given an arbitrary costate sequence $\mathbf{ss}'$, is the product of the probability of the costate sequence and the corresponding similarity of the two state sequences, that is,

$$\begin{aligned} K^*_\rho(\lambda,\lambda'|\mathbf{ss}') &= P(\mathbf{ss}'|\lambda,\lambda')\psi_{\mathbf{ss}'} \\ &= \pi_{s_0}\pi_{s'_0}\prod_{t=1}^{T}a_{s_t|s_{t-1}}a_{s'_t|s'_{t-1}}\prod_{t=0}^{T}\psi_{s_t,s'_t} \\ &= \pi_{s_0}\pi_{s'_0}\psi_{s_0,s'_0}\prod_{t=1}^{T}a_{s_t|s_{t-1}}a_{s'_t|s'_{t-1}}\psi_{s_t,s'_t}. \end{aligned} \tag{11}$$

Then the similarity measure between λ and λ' is obtained by summing $K^*_\rho(\lambda,\lambda'|\mathbf{ss}')$ over all possible costate sequences, that is,

$$\begin{aligned} K^*_\rho(\lambda,\lambda') &= \sum_{\text{all } \mathbf{ss}'} K^*_\rho(\lambda,\lambda'|\mathbf{ss}') \\ &= \sum_{s_0,\dots,s_T=1}^{N}\ \sum_{s'_0,\dots,s'_T=1}^{N'}\pi_{s_0}\pi_{s'_0}\psi_{s_0,s'_0}\prod_{t=1}^{T}a_{s_t|s_{t-1}}a_{s'_t|s'_{t-1}}\psi_{s_t,s'_t}. \end{aligned} \tag{12}$$

From the above perspective, the resulting similarity measure between two Gaussian HMMs is the same as that of (8), but the value of ρ can be chosen freely rather than being confined to 1. From (11), (12), it can be seen that in the new interpretation, the similarity measure of two Gaussian HMMs is calculated as the statistical average of similarities of all possible costate sequences of these two Gaussian HMMs.

The modification of GPPK in this work is developed specifically for Gaussian HMMs, whereas (12) can still be applicable to HMMs with discrete observation distributions or other forms of continuous observation distributions. In these cases, a new similarity measure $\psi_{s,s'}$ for the specific observation distributions needs to be developed. For two Gaussian HMMs, the overall computation complexity for all possible pair of Gaussian states is $O(D^3NN')$. For HMMs with discrete distributed states, this computation is usually much lighter than that of Gaussian states. For HMMs with mixture Gaussian, assuming the numbers of mixtures for each state are N_{m} and N'_{m}, respectively, the overall complexity of all possible pairs of states is $O(D^3N_{\text{m}}N'_{\text{m}}NN')$. If large T is required in the computation, the major computation load still lies in the induction phase rather than in computing the similarities between observation distributions.

3. FORWARD PROCEDURE

The brute-force computation of the similarity measure between two Gaussian HMMs, however, is prohibitively intensive. The computational complexity in the evaluation of the similarity measure under (12) is $O(3T(NN')^{T+1})$. Precisely speaking, there will be $(NN')^{T+1} - 1$ additions and $(NN')^{T+1}(3T - 1)$ multiplications. Clearly a more computational efficient procedure is needed.

In this section, we adopt a forward procedure which is analogous to the popularly used forward procedure in the probability evaluation of the observation sequence on HMMs. First we define the forward similarity measure of two Gaussian HMMs as

$$\begin{aligned} \alpha_\tau(i,j) &= K^*_\rho(\lambda,\lambda', s_\tau = i, s'_\tau = j) \\ &= \sum_{s_0,\dots,s_{\tau-1}}\ \sum_{s'_0,\dots,s'_{\tau-1}} \\ &\quad\times\left(\pi_{s_0}\pi_{s'_0}\psi_{s_0,s'_0}\prod_{t=1}^{\tau-1}a_{s_t|s_{t-1}}a_{s'_t|s'_{t-1}}\psi_{s_t,s'_t}a_{i|s_{\tau-1}}a_{j|s'_{\tau-1}}\psi_{i,j}\right), \end{aligned} \tag{13}$$

that is, the similarity measure of two Gaussian HMMs when only $0 \le t \le \tau$ is considered and $s_\tau = i$, $s'_\tau = j$. Then $\alpha_\tau(i,j)$ can be inductively computed as the following forward procedure.

(1) *Initialization:*

$$\alpha_0(i,j) = \pi_i\pi'_j\psi_{i,j}, \quad 1 \le i \le N, 1 \le j \le N'. \tag{14}$$

(2) *Induction:*

$$\begin{gathered} \alpha_\tau(i,j) = \sum_m\sum_n \alpha_{\tau-1}(m,n)a_{i|m}a_{j|n}\psi_{i,j}, \\ 1 \le \tau \le T,\ 1 \le i \le N, 1 \le j \le N'. \end{gathered} \tag{15}$$

[1] Note that if ρ take values other than $\rho = 1$, it is difficult to compute the GPPK of two HMMs in closed form based on the definition of GPPK.

This is because for $\tau = 1$,

$$\begin{aligned}\alpha_1(i,j) &= \sum_m \sum_n \pi_m \pi'_n \psi_{m,n} a_{i|m} a_{j|n} \psi_{i,j} \\ &= \sum_m \sum_n \alpha_0(m,n) a_{i|m} a_{j|m} \psi_{i,j};\end{aligned} \tag{16}$$

for $\tau > 1$,

$$\begin{aligned}\alpha_\tau(i,j) = &\sum_{s_0,\ldots,s_{\tau-1}} \sum_{s'_0,\ldots,s'_{\tau-1}} \\ &\times \left(\pi_{s_0}\pi_{s'_0}\psi_{s_0,s'_0} \prod_{t=1}^{\tau-1} a_{s_t|s_{t-1}} a_{s'_t|s'_{t-1}} \psi_{s_t,s'_t} a_{i|s_{\tau-1}} a_{j|s'_{\tau-1}} \psi_{i,j}\right) \\ = &\sum_{s_{\tau-1}} \sum_{s'_{\tau-1}} \Bigg(\Bigg(\sum_{s_0,\ldots,s_{\tau-2}} \sum_{s'_0,\ldots,s'_{\tau-2}} \pi_{s_0}\pi_{s'_0}\psi_{s_0,s'_0} \\ &\times \prod_{t=1}^{\tau-2} a_{s_t|s_{t-1}} a_{s'_t|s'_{t-1}} \psi_{s_t,s'_t} a_{s_{\tau-1}|s_{\tau-2}} a_{s'_{\tau-1}|s'_{\tau-2}} \\ &\times \psi_{s_{\tau-1},s'_{\tau-1}} \Bigg) \left(a_{i|s_{\tau-1}} a_{j|s'_{\tau-1}} \psi_{i,j}\right) \Bigg) \\ = &\sum_m \sum_n \alpha_{\tau-1}(m,n) a_{i|m} a_{j|n} \psi_{i,j}.\end{aligned} \tag{17}$$

(3) *Termination:*

$$K^*_\rho(\lambda,\lambda') = \sum_i \sum_j \alpha_T(i,j). \tag{18}$$

The computational complexity of this forward procedure is $O(3T(NN')^2)$. To be precise, there will be $NN'(2+3NN'T)$ multiplications and $NN'(1+(NN'-1)T)$ additions. Comparing to the brute-force computation, the complexity of the forward procedure is much lower especially when T is large.

In (13), the initial state distribution π and the state transition probability distribution a are less than 1. It is apparent that when τ gets big, each term of the sum in (13) goes to zero and the dynamic range of $\alpha_\tau(i,j)$ will go beyond the precision range of any machine. Therefore a scaling procedure is needed to maintain the value of $\alpha_\tau(i,j)$ within the dynamic range of the machine as well as guarantee that the exact value of the similarity measure can be realized.

We denote $\alpha_\tau(i,j)$ as the unscaled forward similarity measure, $\hat{\alpha}_\tau(i,j)$ as the scaled forward similarity measure, and $\hat{\hat{\alpha}}_\tau(i,j)$ as the temporary variable for the computation of $\hat{\alpha}_\tau(i,j)$. Below is the refined forward procedure with the scaling procedure.

(1) *Initialization*. Let $\hat{\hat{\alpha}}_0(i,j) = \alpha_0(i,j)$. Define the scaling coefficient c_0 as $c_0 = (\sum_{i,j} \hat{\hat{\alpha}}_0(i,j))^{-1}$. Let $\hat{\alpha}_0(i,j) = c_0 \hat{\hat{\alpha}}_0(i,j)$.

(2) *Induction*. Let $\hat{\hat{\alpha}}_\tau(i,j) = \sum_m \sum_n \hat{\alpha}_{\tau-1}(m,n) a_{i|m} a_{j|n} \psi_{i,j}$, and $c_\tau = (\sum_{i,j} \hat{\hat{\alpha}}_\tau(i,j))^{-1}$; then $\hat{\alpha}_\tau(i,j) = c_\tau \hat{\hat{\alpha}}_\tau(i,j)$.

(3) *Termination*. From the induction step, it can be found that

$$\hat{\alpha}_\tau(i,j) = c_\tau \hat{\hat{\alpha}}_\tau(i,j) = \cdots = c_\tau c_{\tau-1} \cdots c_0 \alpha_\tau(i,j). \tag{19}$$

Then

$$\begin{aligned}K^*_\rho(\lambda,\lambda') &= \sum_i \sum_j \alpha_T(i,j) \\ &= \frac{1}{c_T c_{T-1} \cdots c_0} \sum_i \sum_j \hat{\alpha}_T(i,j).\end{aligned} \tag{20}$$

Because $K^*_\rho(\lambda,\lambda')$ and $c_T c_{T-1} \cdots c_0$ may also go beyond the dynamic range of the machine, we take the logarithm

$$\log\left(K^*_\rho(\lambda,\lambda')\right) = \log\left(\sum_i \sum_j \hat{\alpha}_T(i,j)\right) - \sum_{t=0}^{T} \log c_t. \tag{21}$$

From the scaling procedure, for each $1 \le \tau \le T$, the values of the scaled αs, $\hat{\alpha}_\tau(i,j)$, are kept within the dynamic range of the computer by multiplying by a scaling coefficient c_τ. By exploiting the relationship between αs and $\hat{\alpha}$s, the exact logarithm value of $K^*_\rho(\lambda,\lambda')$ is realized.

4. FAST SCHEME BASED ON SIMILARITY TRANSITION MATRIX

Comparing to the brute-force computation of the similarity measure between Gaussian HMMs, the forward procedure is computationally efficient. However, this procedure does not consider the time invariant property of state transition matrices of two Gaussian HMMs and their corresponding Gaussian similarity measures.

More specifically, denote the initial probability distribution vector of an HMM as

$$\boldsymbol{\pi} = \begin{bmatrix}\pi_1 & \pi_2 & \cdots & \pi_N\end{bmatrix}^{\mathrm{T}}. \tag{22}$$

Denote the state transition matrix of an HMM as

$$A = \{a_{ij}\} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1N} \\ a_{21} & a_{22} & \cdots & a_{2N} \\ \vdots & \vdots & \vdots & \vdots \\ a_{N1} & a_{N2} & \cdots & a_{NN} \end{bmatrix}. \tag{23}$$

Define the similarity matrix of all possible pair of Gaussians coming from two corresponding Gaussian HMMs, λ and λ', as

$$\boldsymbol{\Psi} = \{\psi_{ij}\} = \begin{bmatrix} \psi_{11} & \psi_{12} & \cdots & \psi_{1N'} \\ \psi_{21} & \psi_{22} & \cdots & \psi_{2N'} \\ \vdots & \vdots & \vdots & \vdots \\ \psi_{N1} & \psi_{N2} & \cdots & \psi_{NN'} \end{bmatrix}. \tag{24}$$

Let $\alpha^0_{ij} = \alpha_0(i,j)$. Define $\theta^{mn}_{ij} = a_{im} a'_{jn} \psi_{mn}$ as the transition similarity measure of two Gaussian HMMs when the state number of these two HMMs are transferred from i to m and from j to n, respectively. Then based on (22)–(24), we define the initial similarity vector and the similarity transition matrix of two Gaussian HMMs as

$$\begin{aligned}\boldsymbol{\alpha}_0 &= \left((\boldsymbol{\pi} \otimes \boldsymbol{\pi}') \odot \operatorname{vec}\left(\boldsymbol{\Psi}^{\mathrm{T}}\right)\right)^{\mathrm{T}} \\ &= \left[\alpha_{11}^0 \ \cdots \ \alpha_{1N'}^0 \ \alpha_{21}^0 \ \cdots \ \alpha_{2N'}^0 \ \cdots \ \alpha_{N1}^0 \ \cdots \ \alpha_{NN'}^0\right],\end{aligned} \tag{25}$$

$$\begin{aligned}\mathbf{S} &= (A \otimes A') \odot \underbrace{\left[\operatorname{vec}(\boldsymbol{\Psi}^{\mathrm{T}}), \operatorname{vec}(\boldsymbol{\Psi}^{\mathrm{T}}), \ldots, \operatorname{vec}(\boldsymbol{\Psi}^{\mathrm{T}})\right]}_{N \times N'}{}^{\mathrm{T}} \\ &= \begin{bmatrix} \theta_{11}^{11} & \cdots & \theta_{11}^{1N'} & \theta_{11}^{21} & \cdots & \theta_{11}^{2N'} & \cdots & \theta_{11}^{N1} & \cdots & \theta_{11}^{NN'} \\ \vdots & \cdots & \vdots & \vdots & \cdots & \vdots & \cdots & \vdots & \cdots & \vdots \\ \theta_{1N'}^{11} & \cdots & \theta_{1N'}^{1N'} & \theta_{1N'}^{21} & \cdots & \theta_{1N'}^{2N'} & \cdots & \theta_{1N'}^{N1} & \cdots & \theta_{1N'}^{NN'} \\ \theta_{21}^{11} & \cdots & \theta_{21}^{1N'} & \theta_{21}^{21} & \cdots & \theta_{21}^{2N'} & \cdots & \theta_{21}^{N1} & \cdots & \theta_{21}^{NN'} \\ \vdots & \cdots & \vdots & \vdots & \cdots & \vdots & \cdots & \vdots & \cdots & \vdots \\ \theta_{2N'}^{11} & \cdots & \theta_{2N'}^{1N'} & \theta_{2N'}^{21} & \cdots & \theta_{2N'}^{2N'} & \cdots & \theta_{2N'}^{N1} & \cdots & \theta_{2N'}^{NN'} \\ \vdots & \cdots & \vdots & \vdots & \cdots & \vdots & \cdots & \vdots & \cdots & \vdots \\ \theta_{N1}^{11} & \cdots & \theta_{N1}^{1N'} & \theta_{N1}^{21} & \cdots & \theta_{N1}^{2N'} & \cdots & \theta_{N1}^{N1} & \cdots & \theta_{N1}^{NN'} \\ \vdots & \cdots & \vdots & \vdots & \cdots & \vdots & \cdots & \vdots & \cdots & \vdots \\ \theta_{NN'}^{11} & \cdots & \theta_{NN'}^{1N'} & \theta_{NN'}^{21} & \cdots & \theta_{NN'}^{2N'} & \cdots & \theta_{NN'}^{N1} & \cdots & \theta_{NN'}^{NN'} \end{bmatrix},\end{aligned} \tag{26}$$

where $\otimes$ is the matrix operator of *Kronecker product*, $\odot$ is the matrix operator of *Hadamard product*, and $\operatorname{vec}(\cdot)$ is the vector operator [10]. The dimension of the initial similarity vector $\boldsymbol{\alpha}_0$ and the similarity transition matrix $\mathbf{S}$ is 1 by $N \times N'$ and $N \times N'$ by $N \times N'$, respectively.

Compare (25)-(26) with (14)–(18), it can be observed that the sum of components of the initial similarity vector $\boldsymbol{\alpha}_0$ is

$$\begin{aligned}&\operatorname{sum}(\boldsymbol{\alpha}_0) \\ &\quad = \operatorname{sum}\left(\left[\alpha_{11}^0 \cdots \alpha_{1N'}^0 \ \alpha_{21}^0 \cdots \alpha_{2N'}^0 \cdots \alpha_{N1}^0 \cdots \alpha_{NN'}^0\right]\right) \\ &\quad = \sum_i \sum_j \alpha_0(i,j),\end{aligned} \tag{27}$$

that is, the similarity of two Gaussian HMMs when $\tau = 0$. It can also be observed that

$$\operatorname{sum}(\boldsymbol{\alpha}_0 \mathbf{S}) = \sum_i \sum_j \alpha_1(i,j), \tag{28}$$

that is, the similarity of two Gaussian HMMs when $\tau = 1$. Likewise, for $\tau = T$, we have

$$\operatorname{sum}\Bigg(\boldsymbol{\alpha}_0 \underbrace{\mathbf{SS}\cdots\mathbf{S}}_{T}\Bigg) = \sum_i \sum_j \alpha_T(i,j) = K_\rho^*(\lambda, \lambda'). \tag{29}$$

Similar to (25), if we define

$$\boldsymbol{\alpha}_\tau = \left[\alpha_{11}^\tau \ \cdots \ \alpha_{1N'}^\tau \ \alpha_{21}^\tau \ \cdots \ \alpha_{2N'}^\tau \ \cdots \ \alpha_{N1}^\tau \ \cdots \ \alpha_{NN'}^\tau\right], \tag{30}$$

where $\alpha_{ij}^\tau = \alpha_\tau(i,j)$, then the forward procedure proposed in the above section can be reinterpreted as the following matrix manipulation.

(1) *Initialization.* Compute $\boldsymbol{\alpha}_0$ and $\mathbf{S}$; see (25)-(26).

(2) *Induction:*

$$\boldsymbol{\alpha}_\tau = \boldsymbol{\alpha}_{\tau-1}\mathbf{S}, \quad 1 \le \tau \le T. \tag{31}$$

(3) *Termination:*

$$K_\rho^*(\lambda, \lambda') = \operatorname{sum}(\boldsymbol{\alpha}_T). \tag{32}$$

From (31), we can see that at the induction step, the transition similarity matrix $\mathbf{S}$ is used iteratively. But the components of the transition similarity matrix include state transition probabilities (a_{ij}s and a'_{ij}s) and GPPKs ($\psi_{ij}s$) corresponding to all possible pairs of Gaussians coming from the two HMMs. They are time invariant with regard to the value of τ. Therefore, it is clear that in order to compute $\boldsymbol{\alpha}_T$, which is needed in the termination step, it is not necessary to follow the induction step of (31). Rather, we can directly compute the T-step transition similarity matrix $\mathbf{S}^{(T)}$, that is,

$$\mathbf{S}^{(T)} = \underbrace{\mathbf{SS}\cdots\mathbf{S}}_{T} = \mathbf{S}^T; \tag{33}$$

and the computation of $\mathbf{S}^{(T)}$ can be accelerated by the following procedure (without loss of generality, we assume $T = 2^n$):

$$\mathbf{S}^{(2^i)} = \mathbf{S}^{(2^{i-1})}\mathbf{S}^{(2^{i-1})}, \quad 1 \le i \le n. \tag{34}$$

So based on the similarity transition matrix, our new proposed procedure is summarized as following:

(1) *Initialization.* Compute $\boldsymbol{\alpha}_0$ and $\mathbf{S}$.

(2) *Induction:*

$$\mathbf{S}^{(2^i)} = \mathbf{S}^{(2^{i-1})}\mathbf{S}^{(2^{i-1})}, \quad 1 \le i \le n; \tag{35}$$

$$\boldsymbol{\alpha}_T = \boldsymbol{\alpha}_0 \mathbf{S}^{(T)}, \quad T = 2^n. \tag{36}$$

(3) *Termination:*

$$K_\rho^*(\lambda, \lambda') = \operatorname{sum}(\boldsymbol{\alpha}_T). \tag{37}$$

The computational complexity of this fast procedure is $O((NN')^3 \log_2 T)$. Specifically, there will be $(NN')^3 \log_2 T + 3(NN')^2$ multiplications and $(NN' - 1)(NN')^2 \log_2 T + (NN')^2$ additions. Comparing to the computational complexity of the forward procedure, $O(3T(NN')^2)$, this fast procedure is advantageous when $T \gg NN'$. So in theory, one can choose one scheme out of the proposed two by determining the value of $3T/NN' \log_2 T$. That is, if $3T/NN' \log_2 T > 1$, the second scheme is preferred. Otherwise, the first scheme is preferred. Because the computational complexity becomes a more critical issue when T is large, the second scheme may have a more practical advantage over the first scheme.

In (35), when i gets big, the values of the components of $\mathbf{S}^{(2^i)}$ will go to zero and rapidly their dynamic range will again go beyond the precision range of any machine. Then another scaling procedure is needed.

Table 1: Model parameter settings of Gaussian HMMs.

Experiment I			Experiment II		
λ_1	λ_2	λ_3	λ_1	λ_2	λ_3
$\boldsymbol{\pi} = [0.6 \quad 0.4]^{\mathrm{T}}$	$\boldsymbol{\pi} = [0.6 \quad 0.4]^{\mathrm{T}}$	$\boldsymbol{\pi} = [0.3 \quad 0.7]^{\mathrm{T}}$	$\boldsymbol{\pi} = [0.6 \quad 0.4]^{\mathrm{T}}$	$\boldsymbol{\pi} = [0.6 \quad 0.4]^{\mathrm{T}}$	$\boldsymbol{\pi} = [0.6 \quad 0.4]^{\mathrm{T}}$
$\mathrm{A}=\begin{bmatrix} 0.4 & 0.6 \\ 0.2 & 0.8 \end{bmatrix}$	$\mathrm{A}=\begin{bmatrix} 0.3 & 0.7 \\ 0.4 & 0.6 \end{bmatrix}$	$\mathrm{A}=\begin{bmatrix} 0.6 & 0.4 \\ 0.3 & 0.7 \end{bmatrix}$	$\mathrm{A}=\begin{bmatrix} 0.3 & 0.7 \\ 0.4 & 0.6 \end{bmatrix}$	$\mathrm{A}=\begin{bmatrix} 0.3 & 0.7 \\ 0.4 & 0.6 \end{bmatrix}$	$\mathrm{A}=\begin{bmatrix} 0.6 & 0.4 \\ 0.8 & 0.2 \end{bmatrix}$
$\boldsymbol{\mu}_1 = [1 \quad 1]^{\mathrm{T}}$	$\boldsymbol{\mu}_1 = [1 \quad 2]^{\mathrm{T}}$	$\boldsymbol{\mu}_1 = [3 \quad 5]^{\mathrm{T}}$	$\boldsymbol{\mu}_1 = [2 \quad 4]^{\mathrm{T}}$	$\boldsymbol{\mu}_1 = [2 \quad 3]^{\mathrm{T}}$	$\boldsymbol{\mu}_1 = [2 \quad 2]^{\mathrm{T}}$
$\boldsymbol{\mu}_2 = [3 \quad 1]^{\mathrm{T}}$	$\boldsymbol{\mu}_2 = [3 \quad 2]^{\mathrm{T}}$	$\boldsymbol{\mu}_2 = [4.5 \quad 3.5]^{\mathrm{T}}$	$\boldsymbol{\mu}_2 = [4 \quad 4]^{\mathrm{T}}$	$\boldsymbol{\mu}_2 = [4 \quad 3]^{\mathrm{T}}$	$\boldsymbol{\mu}_2 = [4 \quad 2]^{\mathrm{T}}$
$\boldsymbol{\Sigma}_1=\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	$\boldsymbol{\Sigma}_1=\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	$\boldsymbol{\Sigma}_1=\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	$\boldsymbol{\Sigma}_1=\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	$\boldsymbol{\Sigma}_1=\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	$\boldsymbol{\Sigma}_1=\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$
$\boldsymbol{\Sigma}_2=\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	$\boldsymbol{\Sigma}_2=\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	$\boldsymbol{\Sigma}_2=\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	$\boldsymbol{\Sigma}_2=\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	$\boldsymbol{\Sigma}_2=\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	$\boldsymbol{\Sigma}_2=\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

We denote $\mathbf{S}^{(2^i)}$ as the unscaled 2^i-step similarity transition matrix, $\hat{\mathbf{S}}^{(2^i)}$ as the scaled 2^i-step similarity transition matrix, and $\hat{\hat{\mathbf{S}}}^{(2^i)}$ as the temporary matrix for the computation of $\hat{\mathbf{S}}^{(2^i)}$. Denote $\hat{\boldsymbol{\alpha}}_T$ as the scaled $\boldsymbol{\alpha}_T$. Below is the refined fast procedure embedded with the scaling procedure (again we assume $T = 2^n$).

(1) *Initialization.* Compute $\boldsymbol{\alpha}_0$ and $\mathbf{S}$. Let $\hat{\hat{\mathbf{S}}}^{(2^0)} = \mathbf{S}$. Let the scaling coefficient c_0 be as $c_0 = (\text{sum}(\hat{\hat{\mathbf{S}}}^{(2^0)}))^{-1}$, where $\text{sum}(\hat{\hat{\mathbf{S}}}^{(2^0)})$ is the sum of all components of $\hat{\hat{\mathbf{S}}}^{(2^0)}$. Let $\hat{\mathbf{S}}^{(2^0)} = c_0\hat{\hat{\mathbf{S}}}^{(2^0)}$.

(2) *Induction.* Let $\hat{\hat{\mathbf{S}}}^{(2^i)} = \hat{\mathbf{S}}^{(2^{i-1})}\hat{\mathbf{S}}^{(2^{i-1})}$, and $c_i = (\text{sum}(\hat{\hat{\mathbf{S}}}^{(2^i)}))^{-1}$; then $\hat{\mathbf{S}}^{(2^i)} = c_i\hat{\hat{\mathbf{S}}}^{(2^i)}$, for $1 \le i \le n$. Hence $\hat{\boldsymbol{\alpha}}_T = \boldsymbol{\alpha}_0\hat{\mathbf{S}}^{(T)}$.

(3) *Termination.* From the induction step, it can be found that

$$\hat{\mathbf{S}}^{(T)} = \hat{\mathbf{S}}^{(2^n)} = c_n\hat{\hat{\mathbf{S}}}^{(2^n)} = \cdots = c_n c_{n-1}^{2^1} \cdots c_0^{2^n} \mathbf{S}^{(2^n)}; \quad (38)$$

then

$$\hat{\boldsymbol{\alpha}}_T = \boldsymbol{\alpha}_0\hat{\mathbf{S}}^{(T)} = \boldsymbol{\alpha}_0 c_n c_{n-1}^{2^1} \cdots c_0^{2^n} \mathbf{S}^{(2^n)} = c_n c_{n-1}^{2^1} \cdots c_0^{2^n} \boldsymbol{\alpha}_T. \quad (39)$$

So the similarity measure $K_\rho^*(\lambda, \lambda')$ is computed as

$$K_\rho^*(\lambda, \lambda') = \text{sum}(\boldsymbol{\alpha}_T) = \left(c_n c_{n-1}^{2^1} \cdots c_0^{2^n}\right)^{-1} \text{sum}(\hat{\boldsymbol{\alpha}}_T). \quad (40)$$

Because $K_\rho^*(\lambda, \lambda')$ and $c_n c_{n-1}^{2^1} \cdots c_0^{2^n}$ will also go beyond the dynamic range of the machine, we need to take the logarithm

$$\log\left(K_\rho^*(\lambda, \lambda')\right) = \log\left(\text{sum}(\hat{\boldsymbol{\alpha}}_T)\right) - \sum_{i=0}^{n} 2^{n-i} \log\left(c_i\right). \quad (41)$$

From the scaling procedure, for each $0 \le i \le n$, the values of the components of the scaled 2^i-step similarity transition matrix $\hat{\mathbf{S}}^{(2^i)}$ are kept within the dynamic range of the computer by multiplying by a scaling coefficient c_i. By exploiting the relationship between the scaled and unscaled similarity transition matrices (38), the exact logarithm value of $K_\rho^*(\lambda, \lambda')$ is realized.

5. EXPERIMENTAL RESULTS

Our experiments include two parts. We first use simulated model parameters to test the effectiveness of the introduced similarity measure of Gaussian HMMs. Secondly we test the effectiveness on a set of real texture images for classification.

5.1. Experiments on simulated model parameters

In this subsection, we perform two experiments. In each experiment, three Gaussian HMMs are used to test their relative similarity measure among each other. The model parameters of the Gaussian HMMs are manually set. In the computation of similarity measures between Gaussians, we set $\rho = 1/2$, that is, Bhattacharyya's measure of affinity between Gaussians [9]. The similarity measure of all possible pairs of Gaussian HMMs among the three Gaussian HMMs are computed for $T = 0, 1, 2, 2^2, \ldots, 2^{10}$. Table 1 lists the model parameters chosen for these two experiments (also depicted in Figure 1).

The setting of simulated model parameters is based on the consideration that it should be easy to make an intuitive judgment of the relative similarity between these models by just looking at the parameters of these models. For example, by looking at Figure 1 and parameters in Table 1, one can intuitively tell that in experiment I, $K_{0.5}(\lambda_1, \lambda_2) > K_{0.5}(\lambda_2, \lambda_3) > K_{0.5}(\lambda_1, \lambda_3)$; and in experiment II, $K_{0.5}(\lambda_1, \lambda_2) > K_{0.5}(\lambda_2, \lambda_3) > K_{0.5}(\lambda_1, \lambda_3)$. Then in experiments I and II, similarities of the HMMs pairs, for example, 1-2, 1-3, and 2-3, are computed with various settings of T, for example, $T = 0, 1, 2, 2^2, \ldots, 2^{10}$. In Figure 2, it can be seen

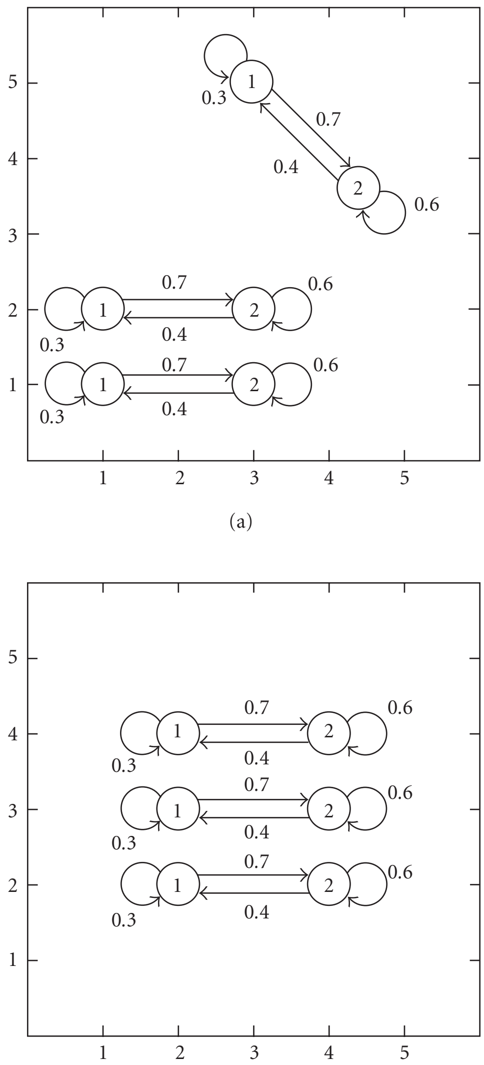


FIGURE 1: Gaussian HMMs used for experiments I and II.

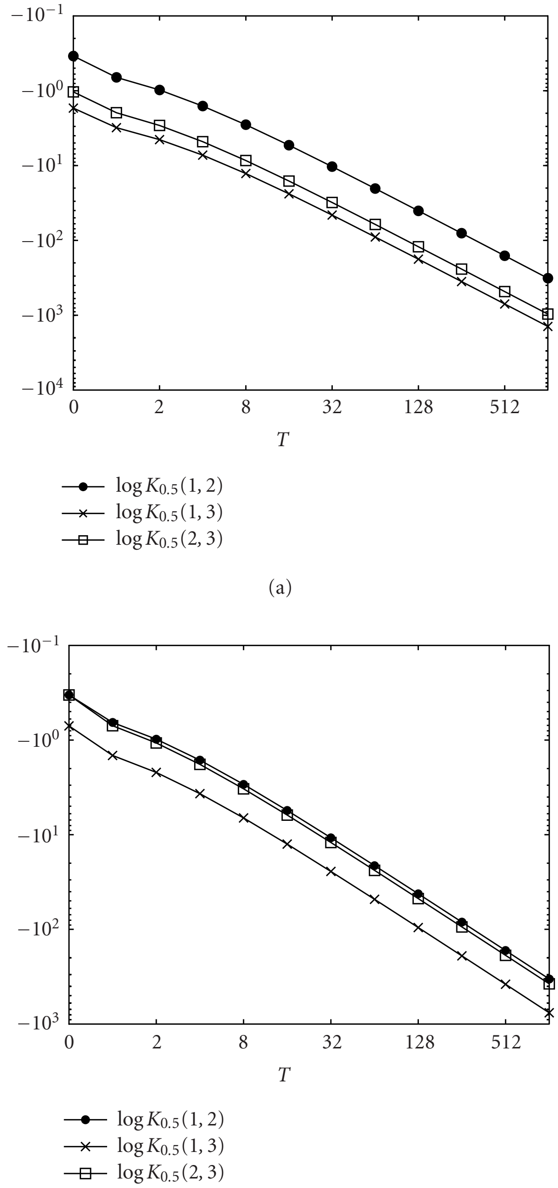


FIGURE 2: (a) and (b) Similarity measures of all possible pairs of Gaussian HMMs in Figures 1a and 1b, respectively.

that not only the similarities of pair 1-2 were always higher than those of pair 1-3 and pair 2-3, but also the magnitudes of the differences among these similarities increase exponentially over T (note that a logarithmic scale was used for the y-axis in Figure 2). This phenomenon suggests that with the increase of T, the proposed similarity measure should become more accurate in classification.

5.2. Experiments on texture classification

In this subsection, the method of similarity measure of Gaussian HMMs is tested on texture classification. thirteen texture images of Brodatz texture images (see the USC-SIPI Image Database at http://sipi.usc.edu/services/database/Database.html) are used for classification; see Figure 3. All 13 texture images are monochrome with size of 512 × 512. Each texture image is divided into 16 128 × 128 nonoverlapping sub-texture images for training and test.

For each sub-texture image, a Gaussian HMM is trained by the observation vector sequence generated from the sub-texture image. The generation of the observation vector sequence can be summarized by using an 8 × 8-sized sliding window to scan the sub-texture image with 75% (or 6 pixels) overlap between consecutive steps from left to right and from top to bottom. The windowed image blocks are normalized to zero mean and further transformed by an 8 × 8 DCT. Only the 3 × 3 lowest frequency coefficients in the DCT domain are used to form the 9-dimensional observation vectors. All consecutive observation vectors form the observation vector sequence.

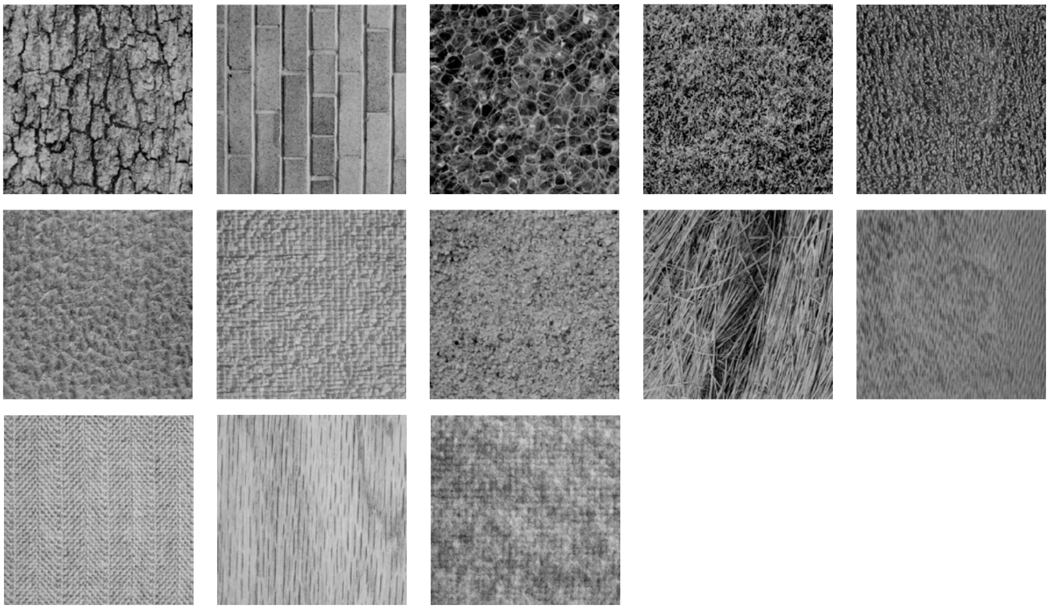


Figure 3: Thirteen categories of texture images. From top to bottom and from left to right, they are bark, brick, bubbles, grass, leather, pigskin, raffia, sand, straw, water, weave, wood, and wool.

5.2.1. Experiment one

In this experiment, for each class of texture, all 16 trained HMMs are selected for each class. Then totally there are $13 \times 16 = 208$ HMMs involved. The similarity measures of all possible pairs of HMMs among all selected 208 HMMs are computed with $T = 4$ and $\rho = 1/2$.[2] Denote $s_{m,n}^{i,j}$ as the log of the similarity measure between the mth HMM of class i and the nth HMM of class j. The similarity measures are arranged by the following similarity measure matrix and depicted in Figure 4a:

$$\begin{array}{ccccccccc} s_{1,1}^{1,1} & s_{1,2}^{1,1} & \cdots & s_{1,16}^{1,1} & \cdots & s_{1,1}^{1,13} & s_{1,2}^{1,13} & \cdots & s_{1,16}^{1,13} \\ s_{2,1}^{1,1} & s_{2,2}^{1,1} & \cdots & s_{2,16}^{1,1} & \cdots & s_{2,1}^{1,13} & s_{2,2}^{1,13} & \cdots & s_{2,16}^{1,13} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ s_{16,1}^{13,1} & s_{16,2}^{13,1} & \cdots & s_{16,16}^{13,1} & \cdots & s_{16,1}^{13,13} & s_{16,2}^{13,13} & \cdots & s_{16,16}^{13,13} \end{array} \tag{42}$$

The brightness of each pixel in Figure 4a represents the value of similarity measure of the corresponding pair of Gaussian HMMs, that is, the brighter the pixel, the greater the similarity measure.

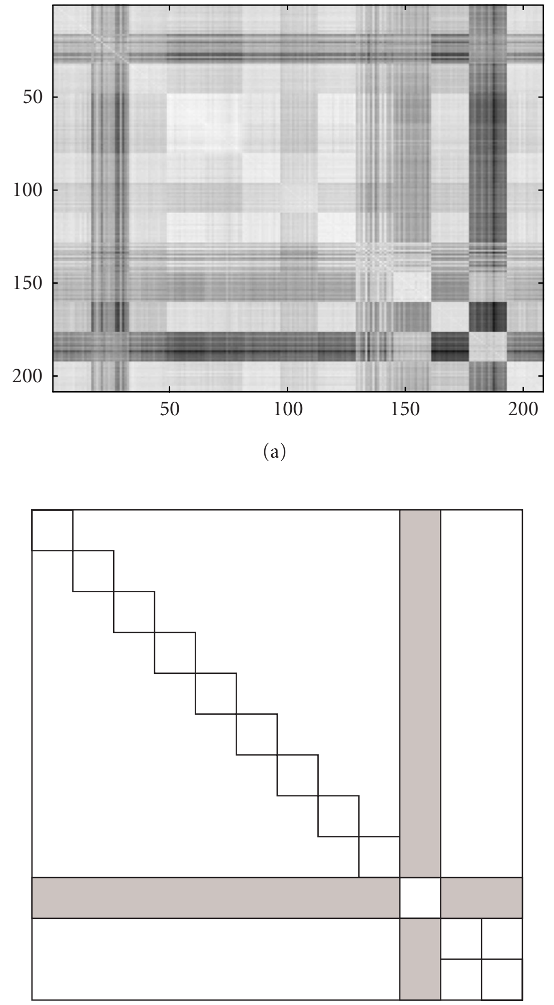
It can be seen from Figure 4a that within-class similarity measures are normally higher than between-class similarity measures, for example, the squares along the diagonal of the similarity measure matrix are generally brighter than the corresponding off-diagonal squares. To illustrate this, Figure 4b shows a sketch of squares along the diagonal of the similarity measure matrix and the corresponding off-diagonal squares (the gray areas).

5.2.2. Experiment two

In this experiment, for each class of textures, 5 trained HMMs are randomly selected. The selected 5 HMMs of each texture class serve as class templates. All the corresponding unselected 11 trained HMMs of each class serve as the testing data. When an arbitrary testing HMM is sent to the classification system, its similarity measures towards all the class templates of every texture class are computed. Then the similarity of the testing HMM towards a particular texture class is computed as the mean value of the similarity measures of the testing HMM towards all the 5 templates of that texture class. The identity of the testing HMM is assigned to the texture class which has the highest similarity measure towards the testing HMM.

For the purpose of observing the convergence property of the similarity measure when T gets big, we tested the recognition rates (the rates according to which the testing HMMs are correctly classified) on $T = 0, 2^0, 2^1, 2^2, \ldots, 2^{10}$; and the ρ is set to be 1/2. When $T = 0$, the classification is actually based on the similarity measure of the observation distributions (Gaussian) of two HMMs and the state transition matrix is not involved in the computation. Obviously, when T gets big, the influence of the state transition matrix in the computation of similarity score gets big. The recognition rate when $T = 0$ is 0.8042. The recognition rates of other settings of T's values are depicted in Figure 5. An interesting observation is that, in this experiment, the recognition rate jumps up from 0.8042 at $T = 0$ to 0.9510 at around $T = 2^2$, and then converges at around 0.90 as T increases. We attribute this phenomenon to the simple Gaussian models used for

[2] We excluded the influence of the initial probability distribution π by substituting all π_i's and π'_j's with $1/N$'s and $1/N'$'s. Due to the limited training data (just one sub-texture image is used in the training of HMM), the initial probability distribution is unreliable and should be excluded from the computation of similarity measures.



(a)

(b)

Figure 4: (a) Similarity measure matrix of 13 × 16 Gaussian HMMs generated from 13 × 16 texture images. (b) Illustrative plot of squares along the diagonal of the similarity measure matrix and the corresponding off-diagonal squares.

HMM state distributions and the inaccuracy in model parameter estimations. With texture images, the assumption of Gaussian model for our DCT domain feature vectors is mostly for computational simplicity; and the estimation of HMM model parameters for subimages from the same texture class may not be consistent due to the variant appearances among these subimages (e.g., note the heterogeneous appearance in the straw image in Figure 3). The suboptimal property of the EM algorithm may also introduce some estimation error. Furthermore, it is reasonable to assume that all three components of HMM model parameters (A, B, π) should share certain influence in determining the similarity measures. However, as stated in the previous subsection, π is excluded from our experiment because of its inaccuracy due to limited training data. Therefore when $T = 0$, the similarity measure is solely determined by B. On the other hand, when $T \to \infty$, A becomes more and more dominant in the computation of similarity score. Therefore over the entire range of T, there may be some point between 0 and ∞ where the combination of contributions from A and B get maximized. Further studies will be conducted to address this phenomenon. When the best T value is not known, a common practice is to set T equal to the length of the observation sequences.

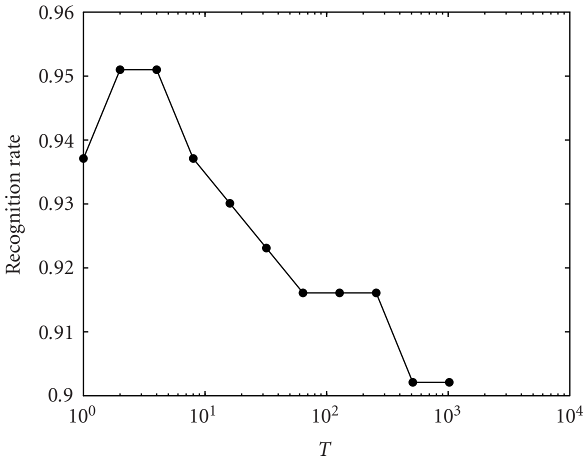


Figure 5: Recognition rates when T is set as $2^0, 2^1, \ldots, 2^{10}$.

6. CONCLUSION

In this work, we introduced a similarity measure of Gaussian HMMs based on a modified "generalized probability product kernel" definition. We also provided a new interpretation for the derivation of this similarity measure. Two fast computing procedures embedded with corresponding scaling procedures were presented. The similarity measure is evaluated on simulated model parameters as well as texture images. Encouraging results testified the effectiveness of the proposed method for similarity comparison between Gaussian HMMs. The method can be further generalized for the comparison of mixture Gaussian HMMs and more complicated stochastic models, and it may also find potential applications in other data analysis areas. The Matlab code for the proposed schemes will be available upon request.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their insightful comments and constructive suggestions.

REFERENCES

[1] F. Samaria, *Face recognition using hidden Markov model*, Ph.D. thesis, University of Cambridge, Cambridge, UK, 1995.

[2] L. Chen, H. Man, and A. Nefian, "Face recognition based on multi-class mapping of Fisher scores," *Pattern Recognition*, vol. 38, no. 6, pp. 799–811, 2005, *Special Issue on Image Understanding for Digital Photographs*.

[3] G. Fan and X.-G. Xia, "Wavelt-based texture analysis and synthesis using hidden Markov models," *IEEE Trans. Circuits Syst. I*, vol. 50, no. 1, pp. 106–120, 2003.
[4] M. N. Do and M. Vetterli, "Rotation invariant texture characterization and retrieval using steerable wavelet-domain hidden Markov models," *IEEE Trans. Multimedia*, vol. 4, no. 4, pp. 517–527, 2002.
[5] B. H. Juang and L. Rabiner, "A probabilistic distance measure for hidden Markov models," *AT&T Technical Journal*, vol. 64, no. 2, pp. 391–408, 1985.
[6] R. B. Lyngsø, C. N. S. Pedersen, and H. Nielsen, "Measures on hidden Markov models," Technical Report RS-99-6, Basic Research in Computer Science, Aarhus, Denmark, 1999.
[7] C. Bahlmann and H. Burkhardt, "Measuring HMM similarity with the Bayes probability of error and its application to online handwriting recognition," in *Proc. IEEE 6th International Conference on Document Analysis and Recognition (ICDAR '01)*, pp. 406–411, Seattle, Wash, USA, September 2001.
[8] T. Jebara and R. Kondor, "Bhattacharyya and expected likelihood kernels," in *Proc. Conference on Learning Theory (COLT '03)*, pp. 57–71, Washington, DC, USA, August 2003.
[9] F. Aherne, N. Thacker, and P. Rockett, "The Bhattacharyya metric as an absolute similarity measure for frequency coded data," *Kybernetika*, vol. 32, no. 4, pp. 1–7, 1997.
[10] J. Schott, *Matrix Analysis for Statistics*, John Wiley & Sons, New York, NY, USA, 1996.

Ling Chen received the B.S. degree from the Northwestern Polytechnical University, China, in 1996, and the M.S. degree from the University of Electronic Science and Technology of China, China, in 1999, both in electrical engineering. Currently, he is a Ph.D. candidate in the Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, New Jersey. His research interests include biometrics, machine learning, statistical pattern recognition, and neural networks.

Hong Man received the B.S. degree from Soochow University, China, in 1988, the M.S. degree from Gonzaga University in 1994, and the Ph.D. degree from Georgia Institute of Technology in 1999, all in electrical engineering. He joined Stevens Institute of Technology in 2000, and currently he is an Assistant Professor in the Department of Electrical and Computer Engineering. He is serving as the Director for Computer Engineering Undergraduate Program in the ECE Department, and the Coordinator for NSA Center of Academic Excellence in Information Assurance in the School of Engineering. He served as a Member of the Organizing Committee for the IEEE International Workshop on Multimedia and Signal Processing (MMSP) 2002 and 2005, Member of the Technical Program Committee for the IEEE Vehicular Technology Conference (VTC) Fall 2003, and the IEEE/ACM International Conference on E-Business and Telecommunication Networks (ICETE) 2004 and 2005. He is a Committee Member in IEEE SPS TC for Education. His research interests include image analysis, medical imaging, multimedia networking, and network security.