GAUSS MIXTURE IMAGE CLASSIFICATION FOR THE LINEAR IMAGE TRANSFORMS

Kivanc M. Ozonat and Robert M. Gray

Information Systems Laboratory
Department of Electrical Engineering
Stanford University, Stanford, CA 94305
{ozonat, rmgray}@stanford.edu

ABSTRACT

Gauss mixture models are commonly used in image classification due to their analytical tractability and robustness. When the feature vectors are formed as the coefficients of a linear image transform, the underlying mixture components are not necessarily Gaussian, in which case there is no guarantee that the Gauss mixture model (GMM)-based clustering algorithms can capture the mixture components. In this work, we train an unbalanced tree-structured GMM-based classifier to reduce this problem. We derive and apply a parameter-independent test to determine the number of mixture components in any given tree node. The classifier tree is grown only in the regions with multiple mixture components.

1. INTRODUCTION

Gauss mixture model-based classifiers are commonly used in image classification due to the analytical tractability and robustness of the Gaussian distribution [1]. When the underlying mixture component distributions are not necessarily Gaussian, however, there is no guarantee that the Gauss mixture model-based clustering algorithms will be able to capture the mixture components. Our work aims to reduce this problem when the feature vector elements are the coefficients of a linear image transform.

The linear image transform coefficients are formed as weighted sums of the image pixels. Accordingly, we invoke the m-dependent central limit theorem and model the marginal pdfs of the mixture components as Gaussians weighted by the distribution of their means and variances. This is an extension of the model introduced by Lam and Goodman [2] and it follows that each component element is a Laplacian, assuming that the block variance follows an exponential distribution.

Based on our model, we first append the image block means and variances to the feature vectors (i.e. vectors of the linear image transform coefficients) so that the marginals of the mixture components are forced to follow a Laplacian distribution. We then derive and apply a parameter-independent test for mixtures of Laplacians to determine whether a given tree node has a single mixture component or multiple mixture components. Our test is an extension of the sign test developed by Lindsay [3] for the one-parameter, two-component exponential mixtures to the two-parameter, multiple-component Laplacian mixtures. We split a tree node only if the test indicates that there is more than a single component in the node.

This work was supported by the National Science Foundation under NSF Grants MIP-9706284-001 and CCR-0073050.

According to our simulations, using a set of aerial images and the discrete cosine transform (DCT) coefficients as feature vector elements, the cross-validated classification error rate of our tree-structured classifier is about 25% lower than the error rates of the full-search classifiers trained using the expectation-maximization (EM) or Lloyd algorithms [1,4].

2. FULL-SEARCH TRAINING BASED ON THE GAUSS MIXTURE MODEL

The Gauss mixture vector quantization (GMVQ)-based classifier is trained using the Lloyd algorithm by iteratively assigning each training vector to the Gauss mixture component minimizing the QDA distortion followed by the update of the parameters of the Gauss mixture components [1]. The QDA distortion due to assigning a training vector $x_{k,l}$ to the component s_i with probability p_i is given by

$$-\ln p_i + \frac{1}{2}\ln\left((2\pi)^D|\Sigma_i|\right) + \frac{1}{2}g_i(x_{k,l}), \tag{1}$$

where $x_{k,l}$ is the D-dimensional feature vector of the image block at location (k,l), μ_i and Σ_i are the mean vector and the covariance matrix of the component s_i, and $g_i(x_{k,l})$ is defined as

$$g_i(x_{k,l}) = (x_{k,l}-\mu_i)^t\Sigma_i^{-1}(x_{k,l}-\mu_i) \tag{2}$$

3. TREE-STRUCTURED GAUSS MIXTURE MODEL-BASED TRAINING

3.1. Tree functionals and pruning

We train a tree-structured classifier, the TS-GMVQ classifier, through growing an unbalanced tree followed by pruning using the BFOS algorithm. The BFOS algorithm requires each node of the tree to have two linear functionals such that one of them is monotonically increasing and the other is monotonically decreasing [5]. Toward this end, we view the QDA distortion of any subtree of the fully-grown tree as a sum of two tree functionals, u_1 and u_2, such that [6,7,8]

$$u_1 = \frac{1}{2}\sum_{s_i\in T} p_i \ln\left((2\pi)^D|\Sigma_i|\right) + \frac{1}{2M}\sum_{s_i\in T}\sum_{x_{k,l}\in s_i} g_i(x_{k,l}), \tag{3}$$

$$u_2 = -\sum_{s_i\in T} p_i \ln p_i, \tag{4}$$

where s_i is the i^{th} node with probability p_i, M is the number of training blocks, T is the set of the terminal tree nodes of the subtree, $x_{k,l}$ is the D-dimensional feature vector of the image block at location (k,l), μ_i and Σ_i are the mean vector and the covariance matrix of node s_i, and $g_i(x_{k,l})$ is defined in (2).

An unbalanced tree is grown by applying the Lloyd algorithm between pairs of children nodes to minimize u_1 after each time a parent node is split.

The splitting stage is followed by pruning based on the BFOS algorithm [5]. By the linearity and monotonicity of the tree functionals, the optimal subtrees (to be pruned) are nested, and at each pruning iteration, the selected subtree is the one that minimizes

$$r = -\frac{\Delta u_1}{\Delta u_2}, \tag{5}$$

where Δu_i, $i = 1, 2$, is the change of the tree functional u_i from the current subtree to the pruned subtree of the current subtree.

The magnitude of (5) increases at each iteration. This is a key point as then, pruning is terminated when the magnitude of (5) reaches 1, resulting in the optimum classification subtree.

3.2. Growing an unbalanced tree

3.2.1. Mixture Modeling

We assume that the pixels of each image block are samples from an m-dependent, stationary process [9]. The m-dependence property can be justified since, in a typical image, the statistical dependence of two sets of pixels, separated by a distance of m pixels both horizontally and vertically, is small for large m. We further assume that the means and variances of the image block pixels are equal to the process mean and variance. From the m-dependent central limit theorem, as the number of samples increases, an appropriately scaled (weighted) sum of samples from a stationary sequence approaches a Gaussian [9]. Thus, when the feature vector elements are the coefficients of a linear image transform, each mixture component element can be viewed as a Gaussian weighted by the distribution of the means and variances of the image blocks assigned to the component.

Using the law of total probability, the underlying distribution, $f(x_d)$ of the d^{th} element, $1 \le d \le D$, of a mixture component can be expressed as

$$f(x_d) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x_d, \mu|\tilde{\sigma}^2) f(\tilde{\sigma}^2)\, d\mu\, d\tilde{\sigma}^2, \tag{6}$$

where $X_d|(\mu, \tilde{\sigma}^2) \sim N(\mu, \sigma^2)$ by the m-dependent central limit theorem, $\mu|\tilde{\sigma}^2 \sim N(\mu_0, \sigma_\mu^2)$ for some mean μ_0 and variance σ_μ^2, and $\tilde{\sigma}^2 = \sigma^2 + \sigma_\mu^2$ (as justified below). Then, (6) reduces to

$$f(x_d) = \int_{-\infty}^{\infty} h(x_d; \mu_0, \tilde{\sigma}^2) f(\tilde{\sigma}^2)\, d\tilde{\sigma}^2, \tag{7}$$

where $h(x_d; \mu_0, \tilde{\sigma}^2)$ is the Gaussian pdf with mean μ_0 and variance $\tilde{\sigma}^2$ evaluated at x_d.

Previous work has shown that the distribution of the image block variances can be approximated well by an exponential distribution [2]. Thus, we let $f(\tilde{\sigma}^2) = \lambda e^{-\lambda(\tilde{\sigma}^2)}$ for some $\lambda > 0$. This justifies setting $\tilde{\sigma}^2 = \sigma^2 + \sigma_\mu^2$ since, by iterated expectation, the image block variance is the sum of σ^2 and σ_μ^2.

Then, using an approach similar to the one used in Lam and Goodman [2], we obtain

$$f(x_d) = \frac{1}{2\lambda'} e^{\frac{-|x_d - \mu_0|}{\lambda'}}, \tag{8}$$

where $\lambda' = \sqrt{1/2\lambda^2}$.

Eq. (8) implies that, when the feature vector elements are the coefficients of a linear image transform, it is reasonable to assume that the mixture component elements are Laplacian. A further implication is that one can force each mixture component element to be a Laplacian by including the block mean and variance in the training stage. Toward this end, we modify the u_1 tree functional to incorporate the distortion due to $\mu_{x_{k,l}}$, the mean, and $\sigma^2_{x_{k,l}}$, the variance, of the image block at location (k,l) as

$$u_1 = \frac{1}{2}\sum_{s_i \in T} p_i \ln\left((2\pi)^D |\Sigma_i|\right) + \frac{1}{2M}\sum_{s_i \in T}\sum_{y_{k,l} \in s_i} g_i(y_{k,l}), \tag{9}$$

where $y_{k,l}$ is the $(D+2)$-dimensional vector consisting of $x_{k,l}$, $\mu_{x_{k,l}}$ and $\sigma^2_{x_{k,l}}$.

Next we derive a parameter-independent test to determine whether a given histogram is that of a single Laplacian pdf or a mixture of multiple Laplacian pdfs.

3.2.2. Multimodal mixtures

A multimodal mixture of Laplacian pdfs is a mixture with multiple local maxima. Any mixture of multiple Laplacians with different means is multimodal. This follows from the piecewise convexity of the Laplacian pdf. Thus, the histogram of a single Laplacian pdf and that of a mixture of multiple Laplacian pdfs with different means can be distinguished by counting the number of maxima in the histograms.

3.2.3. Unimodal mixtures

A unimodal mixture of Laplacian pdfs is a mixture with a single maximum. Any mixture of multiple Laplacians with the same mean is unimodal. This also follows from the piecewise convexity of the Laplacian pdf. Thus, a single maximum suggests either a single Laplacian pdf or a mixture of multiple Laplacian pdfs with the same mean. To distinguish between the two cases, we extend the sign test introduced by Lindsay [3] to a unimodal mixture of multiple Laplacian pdfs.

Theorem : Define the function $R(x)$ as

$$R(x) = p_0 g_0(x) - \sum_{i=1}^{M} p_i g_i(x), \tag{10}$$

such that

$$g_i(x) = \frac{1}{2\lambda_i} e^{-\frac{(|x-\mu|)}{\lambda_i}} \tag{11}$$

$$p_0 = \sum_{i=1}^{M} p_i, \tag{12}$$

$$\lambda_0 = \sqrt{\sum_{i=1}^{M} p_i \lambda_i^2} \tag{13}$$

Then, R(x) has exactly four sign changes in the interval $-\infty < x < \infty$.

Proof. One can express $g_i(x)$ as a function of x and $y = \lambda_i^2$ as

$$h(x,y) = \frac{1}{2\sqrt{y}} e^{-\frac{(|x-\mu|)}{\sqrt{y}}} \tag{14}$$

The second derivative of $h(x,y)$ with respect to y in the interval $\mu < x < \infty$ is given by

$$h^{''}(x,y) = K(x,y)\left(\frac{(x-\mu)^2}{y^{\frac{7}{2}}} - \frac{5(x-\mu)}{y^3} + \frac{3}{y^{\frac{5}{2}}}\right), \tag{15}$$

where $K(x,y) > 0$ for all x and y.

The function $h^{''}(x,y)$ is a quadratic function of x with exactly two sign changes regardless of the value of y in the interval $\mu < x < \infty$. In particular, it takes positive, negative and positive values in the intervals $\mu < x < a_1$, $a_1 < x < a_2$, and $a_2 < x < \infty$, respectively for some a_1 and a_2 with $\mu < a_1 < a_2 < \infty$. Then, by Jensen's inequality, $R(x)$ is negative, positive and negative in the intervals $\mu < x < a_1$, $a_1 < x < a_2$, and $a_2 < x < \infty$, respectively. By symmetry, R(x) has exactly four sign changes in interval $-\infty < x < \infty$. □

The theorem suggests a method to determine whether a unimodal histogram arises from single Laplacian pdf or from a unimodal mixture of multiple Laplacian pdfs. As developed by Lindsay [3], one can count the number of sign changes in $R^{'}(x)$ given by

$$R'(x) = f_1(x) - f_2(x) \tag{16}$$

where $f_1(x)$ is the histogram of a given distribution and $f_2(x)$ is the histogram of the Laplacian with the same mean and variance as the distribution. If $R'(x)$ has four sign changes, this implies that $f_1(x)$ arises from a unimodal mixture of multiple Laplacian pdfs.

3.3. Algorithm

The classifier design algorithm is depicted in Fig. 1. The algorithm starts with a single node tree, called T_1, to which the sign test described below is applied. If the node passes the test, it is split into two child nodes. The Lloyd algorithm is then applied between these two child nodes, minimizing (9), and this new tree is denoted as T_2. Then, the sign test is applied to each terminal node of T_2, and the terminal nodes, which pass the sign test, are split. The Lloyd algorithm is applied between each new pair of child nodes, minimizing (9), to obtain T_3. This procedure of splitting the terminal nodes, which pass the sign test, of T_i, to obtain T_{i+1} and running the Lloyd algorithm between pairs of the child nodes is repeated until $i = L$, where L is pre-selected (Thus, 2^L is analogous to the initial number of components in a full-search algorithm).

Following the tree-growing stage, T_L, the fully-grown tree, is pruned using the BFOS algorithm, until the magnitude of (5) reaches 1.

The Sign Test:

During the tree-growing stage, a node is not split further if the histogram of a feature vector element in the node indicates that the element follows a single Laplacian distribution rather than a mixture distribution. Thus, a node is split only if, for each of the feature vector elements in the node:

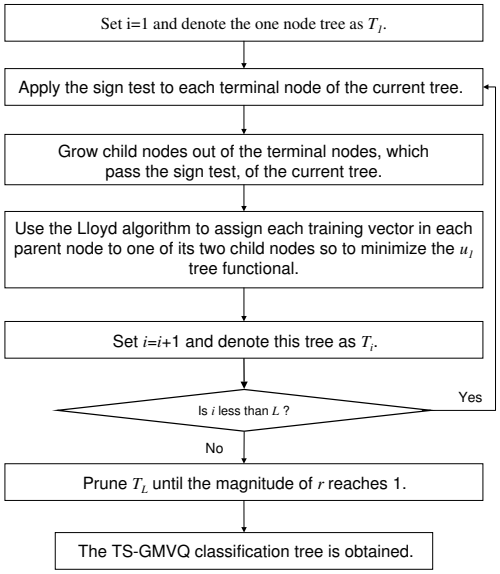

Fig. 1. Classifier Training Algorithm

- There are multiple maxima in the histogram of the feature vector element, or
- There is a single maximum in the histogram of the feature vector element with exactly four sign changes in the difference between the histogram of the feature vector element and the histogram of the Laplacian distribution having the same mean and variance.

4. CLASSIFICATION OF TEST VECTORS

A test vector, $x_{k,l}$, is assigned to the node s_{i^*} if

$$i^* = \arg\min_i \left[-\ln p_i + \frac{1}{2}\ln\left((2\pi)^D|\Sigma_i|\right) + \frac{1}{2}g_i(x_{k,l}) \right] \tag{17}$$

Each node, s_i, is labeled using the majority vote rule. In particular, the number of training vectors of each class, C_j, in node s_i is counted and denoted as $n_{i,j}$ and s_i is labeled as class C_{j^*}, where j^* is given by $j^* = \arg\max_j n_{i,j}$. Any vector assigned to the node s_i is classified by the class label of node s_i.

5. SIMULATIONS AND RESULTS

We used a set of six aerial images in our simulations [10]. Each image is of size 512×512, and is divided into blocks of size 8×8, with each block belonging to one of the two classes. In each experiment, we used 5 of the images for training and the remaining image in the classification stage using cross-validation. The DCT coefficients matrix for each 8×8 image block is computed and the upper-left 4×4 DCT blocks from the DCT coefficients matrices are extracted to be used as feature vectors. Previous work [11] using the same set of images and the GMM-based classifiers has shown that 5 to 8 clusters per class are sufficient to achieve good

classifiers, hence, for each TS-GMVQ experiment, L is selected to be 4 and for each full-search (EM or GMVQ) experiment, the initial number of components is selected to be 16.

Table-1 provides a comparison between the classification error rates of the EM, GMVQ (Lloyd) and TS-GMVQ classifiers using cross-validation. The TS-GMVQ classifier reduces the classification error rate by about 25% from .195 to .149. To have a fair comparison, we did not include the block variances in the test stage for the TS-GMVQ classifier, although they are included in the training stage.

Table-1) Comparison with the Full-Search Algorithms

	EM	GMVQ	TS-GMVQ
Image 1	.206	.212	.171
Image 2	.119	.126	.112
Image 3	.352	.302	.211
Image 4	.256	.246	.220
Image 5	.021	.076	.021
Image 6	.217	.218	.161
Average	.195	.196	.149

Table-2 provides a comparison between three tree-structured classifiers trained in different ways. *Classifier 1* is trained using the u_1 functional as given in (3) and without the sign test (balanced tree), *Classifier 2* is trained using the u_1 functional as given in (9) and without the sign test (balanced tree) and *Classifier 3* (the TS-GMVQ classifier) is trained using the u_1 functional as given in (9) and with the sign test (unbalanced tree). We note that using a balanced tree-structured algorithm (i.e. *Classifier 1*) instead of the full-search algorithms leads to a small improvement in the classification accuracy; most of the improvement is due to using an unbalanced tree-growing strategy with the sign test.

Table-2) Comparison of the Tree-Structured Algorithms

	Classifier 1	Classifier 2	Classifier 3
Image 1	.225	.171	.171
Image 2	.117	.123	.112
Image 3	.278	.211	.211
Image 4	.248	.220	.220
Image 5	.051	.072	.021
Image 6	.201	.161	.161
Average	.185	.160	.149

Table-3 provides a comparison between the classification error rate of the TS-GMVQ classification algorithm and the error rates of the recently published algorithms using the same set of images. These algorithms are described in detail in [10,11,12,13,14]. It should be noted that the TS-GMVQ classifier performs worse than only the classifier based on the HMM-GMM algorithm; however the HMM-GMM algorithm is both a multi-resolution and context-based algorithm.

Table-3) Comparison of Classification Algorithms

Classification Algorithm	Classification Error
HMM-GMM	.140
TS-GMVQ	.149
MHMM	.160
ARM	.178
Causal HMM	.188
Full-search GMVQ	.190
CART	.216
LVQ	.218

6. CONCLUSIONS

We used the central limit theorem and exploited the statistics of the linear image transform coefficients to model the coefficients as Gaussians weighted by their means and variances as an extension of the model proposed by Lam and Goodman [2]. We then derived and applied a parameter-independent sign test, which, based on our model, determines whether a tree node has a single mixture component or multiple mixture components. This test was used to decide which tree nodes should be split further. Our simulations, using the DCT coefficients of a set of aerial images, indicate that our algorithm performs better than the EM and Lloyd algorithms.

7. REFERENCES

[1] R.M. Gray, J.C. Young, and A. K. Aiyer, "Minimum discrimination information clustering: modeling and quantization with Gauss mixtures," *Proc. Int. Conf. Image Processing*, vol. 3, pp. 14-17, 2001.

[2] E.Y. Lam and J.W. Goodman, "Mathematical analysis of the DCT coefficient distributions for images," *IEEE Transactions on Image Processing*, vol. 9, pp. 1661-1666, Oct. 2000.

[3] B.G. Lindsay, *Mixture Models: Theory, Geometry and Applications*, IMS, California, 1995.

[4] A.P. Dempster, N.M. Laird, and D.B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of Royal Statistics Society*, vol. 39, pp. 1-21, 1977.

[5] P.A. Chou, T. Lookabaugh, and R.M. Gray, "Optimal pruning with applications to tree-structured source coding and modeling," *IEEE Transactions on Information Theory*, vol. 35, pp. 299-315, March 1989.

[6] K.M. Ozonat and S. Yoon, "Context-dependent tree-structured image classification using the QDA distortion measure and the hidden Markov model", *Proc. Int. Conf. Image Processing*, Singapore, October 2004, to appear.

[7] K.M. Ozonat, "Image classification using tree-structured discriminant vector quantization," *Proc. Asilomar Conf. on Signals, Systems and Computers*, vol. 2, pp. 1610-1614, 2003.

[8] K.M. Ozonat and R.M. Gray, "Image classification using adaptive boosting and tree-structured discriminant vector quantization", *Proc. Data Compression Conf.*, pp. 556-556, Snowbird, UT, March 2004.

[9] E.L. Lehmann, *Elements of Large-Sample Theory*, Springer-Verlag, New York, 1999.

[10] S. Yoon, C.S. Won, K. Pyun, and R.M. Gray, "Image classification using GMM with context information and reducing dimension for singular covariance," *Proc. Data Compression Conf.*, pp. 457-457, 2003.

[11] K. Pyun, C.S. Won, J. Lim, and R.M. Gray, "Robust image classification based on a non-causal hidden Markov Gauss mixture model," *Proc. Int. Conf. Image Processing*, vol. 3, pp. 785-788, 2002.

[12] J. Li and R.M. Gray, "Image classification by a two-dimensional hidden Markov model," *Proc. Int. Conf. Acoust., Speech and Signal Processing*, vol. 6, pp. 3313-3316, 1999.

[13] J. Li, A. Najmi, and R.M. Gray, "Image classification based on a multi-resolution two-dimensional hidden Markov model," *Proc. Int. Conf. Image Processing*, vol. 1, pp. 348-352, 1999.

[14] A.K. Aiyer, "Robust image compression using Gauss mixture models," Ph.D. dissertation, Stanford University, Department of Electrical Engineering, 2001.