Joint Image Compression and Classification with Vector Quantization and a Two Dimensional Hidden Markov Model

Jia Li Robert M. Gray Richard Olshen
Department of Electrical Engineering
Stanford University, Stanford, CA 94305
Email: jiali@isl.stanford.edu, gray@stanford.edu, olshen@stat.stanford.edu

Abstract

We present an algorithm to achieve good compression and classification for images using vector quantization and a two dimensional hidden Markov model. The feature vectors of image blocks are assumed to be generated by a two dimensional hidden Markov model. We first estimate the parameters of the model, then design a vector quantizer to minimize a weighted sum of compression distortion and classification risk, the latter being defined as the negative of the maximum log likelihood of states and feature vectors. The algorithm is tested on both synthetic data and real image data. The extension to joint progressive compression and classification is discussed.

I Introduction

The issue of joint compression and classification was considered and studied extensively by Oehler, Gray, Perlmutter, Olshen, et al. [13, 14, 16, 17, 15, 18, 19, 11, 6]. A vector quantizer aimed at combining compression and classification generates indices that are mapped into both representative codewords and classes for original vectors at the receiving end. This type of quantization has the potential for several applications in the rapidly growing area of multimedia communication systems. For example, in image databases, to retrieve efficiently a particular image type of interest, it may be required that the codes of images indicate both pixel intensities and image classes. In addition to practical motivation, there is theoretical interest in the study of such vector quantizers. It has been found that the two factors, compression and classification, are not always in conflict; a vector quantizer that minimizes the sum of classification error and a small portion of compression distortion may result in better classification performance than a pure classifier, and vice versa [19, 15].

The joint compression and classification algorithm developed by Oehler, Gray, Perlmutter, Olshen, et al. [13, 14, 16, 17, 15, 18, 19], is referred to as Bayes VQ. The basic assumption is that a training sequence $\mathcal{L} = \{(x_i, y_i), i = 1, 2, ..., L\}$ is a realization of a random process $\{(X_i, Y_i), i = 1, 2, ...\}$ with (X_i, Y_i) obeying a common but unknown distribution P_{XY} on $(X, Y) \in A_X \times A_Y$. Typically P_X is absolutely continuous and is described by a pdf f_X on $\Re^k$, and P_Y is discrete, described by some pmf p_Y. For a testing sequence, we observe $\{x_i, i = 1, 2, ..., L\}$. The goal is to design an encoder $\tilde{\alpha}$, a decoder $\tilde{\beta}$, and a classifier κ to achieve good tradeoff among average distortion, bit rate, and the Bayes risk entailed by guessing Y from the encoded X. The encoder $\tilde{\alpha}$: $A_X \to \mathcal{Z}$, maps x_i to an index m. The decoder $\tilde{\beta}$: $\mathcal{Z} \to A_X$, maps the

index m into a representative vector $\hat{x}_i$ (codeword). The classifier κ: $\mathcal{Z} \rightarrow A_Y$, maps the index into a class. The Bayes VQ algorithm defines a new distortion measure as a weighted sum of compression distortion and penalty for misclassification, which is usually the classification error rate. Vector quantizers are then designed to minimize the average distortion with respect to this measure.

For the Bayes VQ, and many other block based image classification algorithms such as CART$^{\text{TM}}$ [5], images are divided into blocks; and decisions are made independently for the class of each block. In order to improve classification by incorporating context into the decisions regarding classes, a context dependent classification algorithm [10] is developed by modeling images as two dimensional hidden Markov models (HMM). The two dimensional hidden Markov model is extended from the 1-D HMM, that was developed in the 1960s by Baum, Eagon, Petrie, Soules, and Weiss [1, 2, 3, 4]. The basic assumption of a two dimensional hidden Markov model is that, at any block, the image exists in one of a finite set of states. A transition to any state at a block is governed by a fixed probability depending on the states of two adjacent blocks: above and to the left of the current block. Given the state of a block, the feature vector of the block follows a Gaussian distribution with parameters depending on the state. A many to one mapping between states and classes is assumed. Thus, each class may have several states. The classifier first estimates the model based on training data. To classify a test image, the combination of states with the maximum posterior probability given all the feature vectors is searched based on the model. The states are then mapped into classes to obtain the classified image.

The algorithm we describe defines the penalty for misclassification as the negative of the maximum log likelihood of states with feature vectors based on a 2-D HMM. As with the Bayes VQ algorithm, the distortion measure is a weighted sum of compression distortion, in particular, the mean squared error, and the penalty for misclassification. As is discussed in [10], because image blocks are assumed to be statistically dependent, decisions are made jointly for the blocks to classify them optimally. Consequently, blocks with the same feature vector may be classified differently according to their context. A vector quantizer designed with a 2-D HMM has the same property. Hard boundaries between quantization cells vanish due to the context dependent encoding.

In Section II, we introduce notation and define the distortion measure. Section III describes the optimality properties of a vector quantizer with a 2-D HMM and the design algorithm. Details on optimal encoding are provided in Section IV. Section V presents simulation results on both synthetic data and real image data. In Section VI, we discuss the extension of the algorithm to perform joint progressive compression and classification by using 2-D multiresolution HMMs extended from 2-D HMMs.

II Distortion Measure

A training sequence of image data is represented by $\mathcal{L} = \{(x_{k,l}, y_{k,l}), (k,l) \in \mathbb{N}\}$, where $\mathbb{N} = \{(k,l); 1 \leq k \leq K, 1 \leq l \leq L\}$ denotes the collection of blocks in the training image. The random vector $X_{k,l}$ contains the intensities of pixels in block (k,l), and $Y_{k,l}$ is the class of the block. The domain of $X_{k,l}$ is A_X, and the domain

of $Y_{k,l}$ is A_Y. We assume that there exists a random process $\{(u_{k,l}, s_{k,l}), (k,l) \in \mathbb{N}\}$ associated with the training image. The random vector $U_{k,l}$ is the feature vector for block (k,l), which is a function of $X_{k,l}$. The random variable $S_{k,l}$ is the underlying state of the block. It is assumed that $\{(U_{k,l}, S_{k,l}), k,l = 1,2,...\}$ is generated by a 2-D HMM. Details about the assumptions of a 2-D HMM is in [10].

As the state process $\{s_{k,l}, (k,l) \in \mathbb{N}\}$ can never be observed, the 2-D HMM is estimated based on $\{(u_{k,l}, y_{k,l}), (k,l) \in \mathbb{N}\}$ obtained from the training image. We assume that the 2-D HMM is already estimated from the training data. Readers are referred to [10] for details about estimating the parameters of the model. In order to use the 2-D HMM to classify images, feature vectors $\{u_{k,l}, (k,l) \in \mathbb{N}\}$ are evaluated from pixel intensity vectors $\{x_{k,l}, (k,l) \in \mathbb{N}\}$. We then apply the model to search for the states $\{s_{k,l}, (k,l) \in \mathbb{N}\}$ that yield the maximum posterior probability given the feature vectors, i.e., $\max^{-1}_{s_{k,l}} P\{s_{k,l}, (k,l) \in \mathbb{N} \,|\, u_{k,l}, (k,l) \in \mathbb{N}\}$, which is equivalent to the maximization of the joint likelihood with the feature vectors $\max^{-1}_{s_{k,l}} P\{s_{k,l}, u_{k,l}, (k,l) \in \mathbb{N}\}$. The classes are then mapped from the states. We denote the mapping from states to classes by $C(s_{k,l})$. Although we cannot claim in general that the states with the maximum posterior probability necessarily yield the classes with the maximum posterior probability given the feature vectors, we use the likelihood of the states with the feature vectors, $P\{s_{k,l}, u_{k,l}, (k,l) \in \mathbb{N}\}$, as an indication of classification risk because the evaluation of $P\{y_{k,l}, u_{k,l}, (k,l) \in \mathbb{N}\}$ is computationally too intensive. It is reasonable to make such a replacement due to the belief that the optimal decision on the states should yield a good decision on the classes. As a result, for the encoder $\tilde{\alpha}$, given $\hat{y}_{k,l}$, the estimation of respective classes $y_{k,l}$, the penalty for misclassification is $-\max_{s_{k,l}:C(s_{k,l})=\hat{y}_{k,l}} \log(P\{s_{k,l}, u_{k,l}, (k,l) \in \mathbb{N}\})$, i.e., the negative of the maximum log likelihood of the feature vectors and states that can be mapped into classes $\hat{y}_{k,l}$. For the classifier at the receiving end, to decide the optimal class of an index, the goal is to minimize the classification error rather than the error rate of the states.

We define our distortion measure as a Lagrangian function formed in the manner of [7, 19]. Suppose $x_{k,l}$ is encoded as $i_{k,l}$. In our study, the compression distortion $d(x_{k,l}, \tilde{\beta}(i_{k,l}))$ is assumed to be the mean squared error between $x_{k,l}$ and the codeword to which it is decoded . The Lagrangian distortion between an input image $\{x_{k,l}, (k,l) \in \mathbb{N}\}$ and encoder output indices $\{i_{k,l}, (k,l) \in \mathbb{N}\}$ is defined as

$$\rho_\lambda = \frac{1}{KL}[\sum_{(k,l)\in\mathbb{N}} d(x_{k,l}, \tilde{\beta}(i_{k,l})) - \lambda \max_{s_{k,l}:C(s_{k,l})=\kappa(i_{k,l})} \log(P\{s_{k,l}, u_{k,l}, (k,l) \in \mathbb{N}\})] \quad ,$$

where KL is the number of blocks in the image.

The interaction between compression and classification is not as obvious as with Bayes VQ [8, 16] because the encoder and the classifier affect each other indirectly through the choice of states. For each $x_{k,l}$ encoded as $i_{k,l}$, its class $\kappa(i_{k,l})$ determined by the classifier restricts the possible states for block (k,l) to be those satisfying $C(s_{k,l}) = \kappa(i_{k,l})$. We have defined the Lagrangian distortion for the entire image because the likelihood cannot be separated into independent items for each block. Usually we do not encode an entire image jointly; instead, we simplify by dividing the image into subimages that contain a set of image blocks. The distortion ρ_λ is the

distortion for one subimage. The expected value of ρ_λ quantifies the performance of the vector quantizer. We define the expected distortion for the vector quantizer as in [8]:

$$\begin{aligned} J_\lambda(\tilde{\alpha}, \tilde{\beta}, \kappa) &= E(\rho_\lambda) = D(\tilde{\alpha}, \tilde{\beta}) + \lambda B(\tilde{\alpha}, \kappa) \ ; \\ D(\tilde{\alpha}, \tilde{\beta}) &= E(d(X, \tilde{\alpha}(X)) \ ; \\ B(\tilde{\alpha}, \kappa) &= -\frac{1}{KL} E[\max_{s_{k,l}: C(s_{k,l}) = \kappa(i_{k,l})} \log(P\{s_{k,l}, u_{k,l}, (k,l) \in \mathbb{N}\})] \ . \end{aligned}$$

III Optimality Properties and the Algorithm

As with Bayes VQ [8], there are necessary conditions for overall optimality of a vector quantizer with a 2-D HMM. They lead to an iterative algorithm for designing the quantizer. The conditions for the optimal decoder and the optimal classifier are the same as those stated in [8] for the particular case of classification error rate being the penalty for misclassification.

- Given $\tilde{\alpha}$ and κ, the optimal decoder is $\tilde{\beta}(i) = \min_{z \in \hat{A}_X}{}^{-1} E[d(x,z) \,|\, \tilde{\alpha}(X) = i]$.
- Given $\tilde{\alpha}$ and $\tilde{\beta}$, the optimal classifier is $\kappa(i) = \max_k{}^{-1} \hat{P}(Y = k \,|\, \tilde{\alpha}(X) = i)$, that is, a majority vote on the classes of all the vectors encoded to the index. The $\hat{P}(\cdot)$ denotes the empirical frequency for a class based on all the vectors encoded to the index.
- Given κ and $\tilde{\beta}$, then the optimal encoder is

$$\begin{aligned} \tilde{\alpha}(x_{k,l}; (k,l) \in \mathbb{N}) = \min_{i_{k,l}}{}^{-1} \{ & \sum_{(k,l) \in \mathbb{N}} d(x_{k,l}, \tilde{\beta}(i_{k,l})) - \\ & \lambda \cdot \max_{s_{k,l}: C(s_{k,l}) = \kappa(i_{k,l})} \log P\{s_{k,l}, u_{k,l};\ (k,l) \in \mathbb{N}\} \} \ . \end{aligned}$$

The algorithm iterates the three steps in succession. Since we cannot claim the penalty for misclassification used in the encoder is a increasing function of the classification error rate, the algorithm is not guaranteed to be descending. However, simulations that we have performed on both synthetic data and real image data with a wide range of λ have always provided descending Lagrangian distortions.

To design the initial quantizer, we first apply a pure classification with the 2-D HMM on the training data. Based on the classification result, we design a codebook for each class using standard Lloyd algorithm. Bit allocation is used to decide the initial number of codewords for each class. The combination of all the codebooks forms the initial decoder. The classes for the codebooks form the initial classifier.

IV Optimal Encoding

According to the iterative algorithm presented in the previous section, it is simple to update the decoder and the classifier. Recall that the optimal encoder is

$$\begin{aligned}\tilde{\alpha}(x_{k,l}; (k,l) \in \mathbb{N}) &= \min_{i_{k,l}}{}^{-1}\{ \sum_{(k,l)\in\mathbb{N}} d(x_{k,l}, \tilde{\beta}(i_{k,l})) - \\ &\qquad \lambda \cdot \max_{s_{k,l}:C(s_{k,l})=\kappa(i_{k,l})} \log P\{s_{k,l}, u_{k,l};\ (k,l) \in \mathbb{N}\}\} \ .\end{aligned}$$

Consider

$$\begin{aligned}&\min_{i_{k,l}}\{ \sum_{(k,l)\in\mathbb{N}} d(x_{k,l}, \tilde{\beta}(i_{k,l})) - \lambda \cdot \max_{s_{k,l}:C(s_{k,l})=\kappa(i_{k,l})} \log P\{s_{k,l}, u_{k,l};\ (k,l) \in \mathbb{N}\}\} \\ = \ & -\max_{s_{k,l}} \max_{i_{k,l}:\kappa(i_{k,l})=C(s_{k,l})} [\lambda \log P\{s_{k,l}, u_{k,l};\ (k,l) \in \mathbb{N}\} - \sum_{(k,l)\in\mathbb{N}} d(x_{k,l}, \tilde{\beta}(i_{k,l}))]\end{aligned}$$

Since for fixed $s_{k,l}$, $(k,l) \in \mathbb{N}$,

$$\begin{aligned}&\max_{i_{k,l}:\kappa(i_{k,l})=C(s_{k,l})} [\lambda \log P\{s_{k,l}, u_{k,l};\ (k,l) \in \mathbb{N}\} - \sum_{(k,l)\in\mathbb{N}} d(x_{k,l}, \tilde{\beta}(i_{k,l}))] \\ = \ & \lambda \log P\{s_{k,l}, u_{k,l};\ (k,l) \in \mathbb{N}\} - \sum_{(k,l)\in\mathbb{N}} \min_{i_{k,l}:\kappa(i_{k,l})=C(s_{k,l})} d(x_{k,l}, \tilde{\beta}(i_{k,l})) \quad ,\end{aligned}$$

and $\min_{i_{k,l}:\kappa(i_{k,l})=C(s_{k,l})} d(x_{k,l}, \tilde{\beta}(i_{k,l}))$ is simply the minimum mean squared error with a codeword that is classified as $C(s_{k,l})$, the critical step is thus to find $\{s_{k,l}\}$ that maximize

$$\lambda \log P\{s_{k,l}, u_{k,l};\ (k,l) \in \mathbb{N}\} - \sum_{(k,l)\in\mathbb{N}} \min_{i_{k,l}:\kappa(i_{k,l})=C(s_{k,l})} d(x_{k,l}, \tilde{\beta}(i_{k,l})) \quad .$$

The first item is what an image classifier based on a 2-D HMM normally maximizes [10]. Let T_m denote the sequence of states on diagonal m, i.e., $T_m = \{s_{k,l}, (k,l) : k+l = m\}$, as shown in [10], and $m = 0, 1, ..., w+z-2$, where w is the number of rows and z is the number of columns. It can be shown that

$$\begin{aligned}&\lambda \log P\{s_{k,l}, u_{k,l};\ (k,l) \in \mathbb{N}\} - \sum_{(k,l)\in\mathbb{N}} \min_{i_{k,l}:\kappa(i_{k,l})=C(s_{k,l})} d(x_{k,l}, \tilde{\beta}(i_{k,l})) \\ = \ & \sum_{m=0}^{w+z-2} [\lambda \log(P(T_m \,|\, T_{m-1}) P(\mathbf{u}_m \,|\, T_m)) - \sum_{(k,l):k+l=m} \min_{i_{k,l}:\kappa(i_{k,l})=C(s_{k,l})} d(x_{k,l}, \tilde{\beta}(i_{k,l}))] \ . \quad (1)\end{aligned}$$

Equation (1) demonstrates the one-step memory of the Lagrangian distortion in terms of T_m; we can apply the Viterbi algorithm to find the optimal $s_{k,l}$. This is the same method used to search for the optimal states by a pure classifier based on a 2-D HMM. The only difference caused by the compression distortion is an extra cost at each step of the Viterbi transition diagram. A detailed discussion of the Viterbi algorithm as it applies to minimize $\log P\{s_{k,l}, u_{k,l};\ (k,l) \in \mathbb{N}\}$ is in [10].

V Examples

V.1 Synthetic Data

We simulated the algorithm on an extended form of Kohonen Gaussian mixture source [9]. As with an ordinary Kohonen Gaussian mixture, there are two classes. Given the class k, $k = 0, 1$, the random vector X is Gaussian $\mathcal{N}(0, \sigma_k^2)$. In particular, $\sigma_0^2 = 1$,$\sigma_1^2 = 4$. An ordinary Kohonen Gaussian mixture assumes that classes of blocks are iid with equal probabilities for class 0 and 1. We assume, however, that the classes are produced by a 2-D HMM. In this case, the classes are the same as the underlying states because we only assign one state to each class. We also assume that the feature vectors are the same as the vectors to be encoded, i.e., x. The specific transition probabilities are as follows

$$a_{q,n,r} = 0.5, \text{ if } q \neq n; \quad a_{q,n,r} = 0.8, \text{ if } q = n = r; \quad a_{q,n,r} = 0.2, \text{ if } q = n \neq r; \quad ,$$

that is, if the classes of two adjacent blocks are different, then the transition probabilities are the same for both classes. If the classes of two adjacent blocks are consistent, the probability of remaining in the same state is higher than that to move to the different state.

λ	MSE	P_e
0.1	0.601	0.265
1	0.617	0.251
5	0.722	0.240
10	0.746	0.238

Algorithms	MSE	P_e
BVQ: Inverse halftone estimator	0.655	0.269
BVQ: CART-based estimator	0.653	0.274
Cascade	0.598	0.295
BVQ: TSVQ pmf estimator	0.630	0.270

Table 1: Left: MSE and P_e for Kohonen's Example of vector quantizers with 2-D HMM. Right: MSE and P_e for Kohonen's Example of several algorithms.

Using this model, we generated a training data set of size 256×256. We then used the training data to estimate a 2-D HMM. Based on the estimated HMM, we have designed vector quantizers with different distortion weights λ. A test set of size 128×128 is generated to evaluate the performance of the quantizers. If the dependence among blocks is ignored, the source we are considering is simply an ordinary Kohonen Gaussian mixture. The Bayes decision rule yields an error probability of 0.264. Our classifier with the 2-D HMM obtains an error rate of 0.243 for the test data. We have decreased the error rate of the Bayes rule by exploiting the dependency among vectors. The error rate of the Bayes rule is only a lower bound for the performance of classifiers that make decisions independently on each block. At 1.5 bpp, the compression and classification performance versus λ are listed in the left part in Table 1. We see the tradeoff between compression and classification with variable λ. At $\lambda = 5, 10$, the vector quantizer achieves better classification than a pure classifier based on the HMM. Results at the same bit rate for Bayes VQ with different density estimators are provided in [8]. We list some of them in the right part in Table 1. By taking advantage of the inter-block dependencies, the vector quantizer with the 2-D HMM improves both compression and classification.

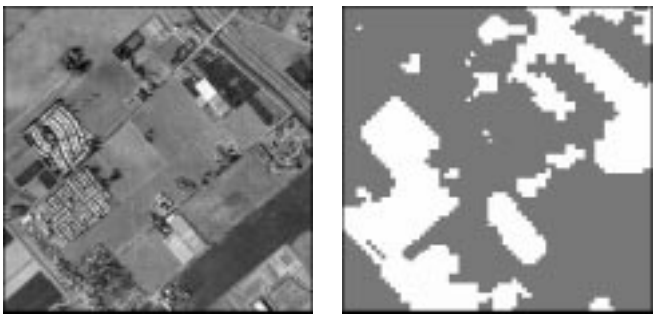

Figure 1: An image and its hand labeled classes. White: man-made, Gray: natural.

V.2 Image Data

Aerial images were segmented into man-made and natural regions as is shown in Figure 1. The images are 512×512 gray-scale images with 8 bits per pixel. They are the aerial images of the San Francisco Bay area provided by TRW (formerly ESL, Inc.) [12]. The application of BVQ to aerial images has been discussed in [12, 19, 15]. Four images were used to train a model and quantizers. The test image is the one shown in Figure 1.

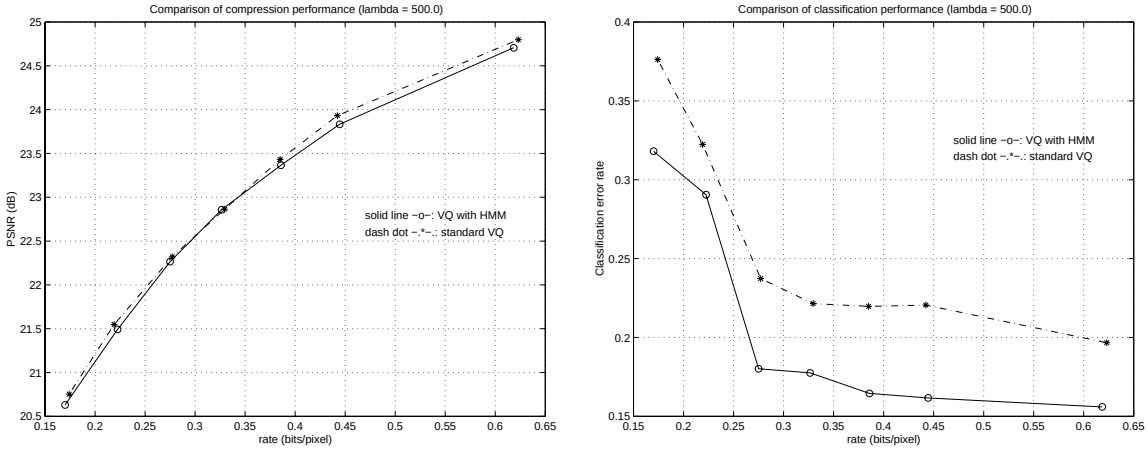

Figure 2: Compression and classification performance at $\lambda = 500.0$.

We divided the images into 4×4 blocks and used DCT coefficients or averages of some of them as features. Denote the DCT coefficients for a 4×4 block by $\{D_{i,j}, i, j \in (0, 1, 2, 3)\}$. The features used are $f_1 = D_{0,0}$, $f_2 = |D_{1,0}|$, $f_3 = |D_{0,1}|$, $f_4 = \frac{\sum_{i=2}^{3}\sum_{j=0}^{1}|D_{i,j}|}{4}$, $f_5 = \frac{\sum_{i=0}^{1}\sum_{j=2}^{3}|D_{i,j}|}{4}$, and $f_6 = \frac{\sum_{i=2}^{3}\sum_{j=2}^{3}|D_{i,j}|}{4}$. The vectors $x_{k,l}$ are 16 dimensional, with each component being the intensity of a pixel. For $\lambda = 500.0$, compression and classification as they vary with bit rate are shown in Figure 2. Performance is compared with a cascade system with standard Lloyd vector quantizer followed by a classifier. The VQ with the 2-D HMM achieves much lower classification error rates while keeping the PSNR about 0.05dB worse (sometimes even better) than that of the cascaded system. In [15], the same image data set is used to test BVQ. Performance is evaluated by cross-validation. At 0.525 bits per pixel, BVQ achieves

about 24.1dB PSNR and about 21% classification error rate. To show the effect of λ on the tradeoff between compression and classification, we present the performance for a variety of λ's at three bit rates in Figure 3. The λ's at the experiment points, ranging from 10 to 10^5, are in an increasing order with the increase of PSNR. As is shown in the figure, some tradeoff with compression can actually improve the classification performance. This is consistent with comments in [5] to the effect that to be greedy for reduction in Bayes risk alone leads to poor classifiers. The differences of the compression distortion between the rates at all the λ are about the same. For λ below 1000, the classification error rate decreases with the bit rate. But for λ above 1000, the classification error rate converges, and higher bit rate does not necessarily lead to lower classification error rate.

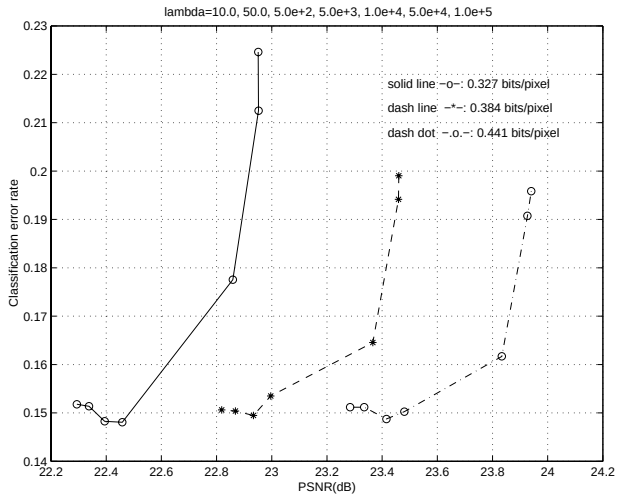

Figure 3: Tradeoff between compression and classification at rates: 0.327bpp, 0.384bpp, and 0.441bpp. The λ's are in an increasing order with the increase of PSNR; and $\lambda = 10.0, 50.0, 5.0 \times 10^2, 5.0 \times 10^3, 1.0 \times 10^4, 5.0 \times 10^4, 1.0 \times 10^5$.

VI Extension to Joint Progressive Compression and Classification

As we can extend the 2-D hidden Markov model for image classification to a multiresolution model in which it is assumed that an image is represented by feature vectors across several resolutions. Images at different resolutions are first obtained using wavelet transforms or other filtering techniques. The feature vectors at a particular resolution are then evaluated based only on the image at that resolution. At each resolution, the model is similar to an HMM except that transition probabilities depend on both adjacent blocks and blocks at the same spatial location in the previous resolution. Since one block at a lower resolution corresponds to several blocks at the same spatial location in the higher resolution (e.g., a quadtree structure), the block in the lower resolution may contain several classes. Consequently, except for the highest

resolution, there exists an extra "mixed" class besides the original classes. Based on the multiresolution model, the classification is progressive in the sense that, at low resolutions, some blocks are classified and the others are marked as the "mixed" class and are subdivided in higher resolutions with a class assigned to each subblock separately. By incorporating the multiresolution model with a progressive vector quantizer, we can perform progressive compression and classification jointly.

Acknowledgments

The authors gratefully acknowledge the helpful comments of the reviewers, which improved the clarity of the paper.

References

[1] L. E. Baum, "An Inequality and Associated Maximization Technique in Statistical Estimation for Probabilistic Functions of Finite State Markov Chains," *Inequalities III*, pp. 1-8, Academic Press, New York, 1972.

[2] L. E. Baum and J. A. Eagon, "An Inequality with Applications to Statistical Estimation for Probabilistic Functions of Markov Processes and to a Model for Ecology," *Bulletin of American Math. Stat.*, pp. 360-363, Vol. 37, 1967.

[3] L. E. Baum and T. Petrie, "Statistical Inference for Probabilistic Functions of Finite State Markov Chains," *Annals of Math. Stat.*, pp. 1554-63, Vol. 37, 1966.

[4] 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.*, pp. 164-171, Vol. 41, No. 1, 1970.

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

[6] N. Chaddha, K. Perlmutter, and R. M. Gray, "Joint Image Classification and Compression Using Hierarchical Table-lookup Vector Quantization," *Proc. Data Compression Conference (DCC)*, pp. 23-32, Snowbird, Utah, March 1996.

[7] P. A. Chou, T. Lookabaugh, and R. M. Gray, "Entropy-constrained Vector Quantization," *IEEE Trans. Acoust. Speech & Sig. Proc.*, vol. 37, pp. 31-42, Jan. 1989.

[8] R. M. Gray, K. O. Perlmutter, and R. A. Olshen, "Quantization, Classification, and Density Estimation for Kohonen's Gaussian Mixture," *Proc. Data Compression Conference*, pp. 63-72, Snowbird, Utah, March 1998.

[9] T. Kohonen, G. Barna, and R. Chrisley, "Statistical Pattern Recognition with Neural Networks: Benchmarking Studies," *IEEE International Conference on Neural Networks*, pp. I-61-68, July 1988.

[10] J. Li, A. Najmi and R. M. Gray, "Image Classification by the Two Dimensional Hidden Markov Model," *Proc. IEEE Int. on Conf. Acoust. Speech & Sig. Proc.*, to appear, Arizona, March 1999.

[11] C. L. Nash, K. O. Perlmutter, and R. M. Gray, "Evaluation of Bayes Risk Weighted Vector Quantization with Posterior Estimation in the Detection of Lesions in Digitized Mammograms," *Proc. of the 28th Asilomar Conf. on Circuits Systems and Computers*, vol. 1, pp. 716-20, Pacific Grove, CA, Oct. 1994.

[12] K. L. Oehler, "Image Compression and Classification Using Vector Quantization," *Ph.D thesis*, Stanford University, 1993.

[13] K. L. Oehler and R. M. Gray, "Combining Image Classification and Image Compression Using Vector Quantization," *Proc. Data Compression Conference*, pp. 2-11, Snowbird, Utah, March 1993.

[14] K. L. Oehler and R. M. Gray, "Combining Image Compression and Classification Using Vector Quantization," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 17, pp. 461-73, May 1995.

[15] K. O. Perlmutter, "Compression and Classification of Images Using Vector Quantization and Decision Trees," *Ph.D thesis*, Stanford University, 1995.

[16] K. O. Perlmutter, R. M. Gray, K. L. Oehler, and R. A. Olshen, "Bayes Risk Weighted Tree-structured Vector Quantization with Posterior Estimation," *Proc. Data Compression Conference*, pp. 274-83, IEEE Computer Society Press, March 1994.

[17] K. O. Perlmutter, R. M. Gray, R. A. Olshen, and S. M. Perlmutter, "Bayes Risk Weighted Vector Quantization with CART Estimated Posteriors", *Proc. IEEE Int. Conf. on Acoust. Speech & Sig. Proc.*, vol. 4, pp. 2435-8, May 1995.

[18] K. O. Perlmutter, C. L. Nash, and R. M. Gray, "A Comparison of Bayes Risk Weighted Vector Quantization with Posterior Estimation with Other VQ-based Classifiers," *Proc. IEEE Int. Conf. on Image Proc.*, vol. 2, pp. 217-21, Austin, TX, Nov. 1994.

[19] K. O. Perlmutter, S. M. Perlmutter, R. M. Gray, R. A. Olshen, and K. L. Oehler, "Bayes Risk Weighted Vector Quantization with Posterior Estimation for Image Compression and Classification," *IEEE Transactions on Image Processing*, vol. 5, no. 2, pp. 347-60, February 1996.