Classification of Features and Images using Gauss Mixtures with VQ Clustering [1]

Ying-zong Huang, Deirdre B. O'Brien & Robert M. Gray
Dept. of Electrical Engineering
Stanford University
Stanford, CA 94305
{zong, dbobrien, rmgray}@stanford.edu

Abstract

Gauss mixture (GM) models are frequently used for their ability to well approximate many densities and for their tractability to analysis. We propose new classification methods built on GM clustering algorithms more often studied and used for vector quantization (VQ). One of our methods is an extension of the 'codebook matching' idea to the specific case of classifying whole images. We apply these methods to a realistic supervised classification problem and empirically evaluate their performances compared with other classification methods.

1 Introduction

Gauss mixture (GM) models have long been used to estimate arbitrary probability densities, especially densities that can be considered as mixtures of several modes. Historically, GM modeling played a fundamental role in the development of speech compression systems (e.g. LPC). More generally, the performance and robustness properties of GM models have been extensively analyzed within the framework of building classified vector quantizers [1].

We use GM models instead to build classifiers for a dataset. From a set of class-labeled training data, we can learn the underlying distributions of the sources for the various classes by training GM models to the given data as if designing quantizers, by means of GM clustering algorithms. Density estimates thus obtained can be used to make classification decisions. In instances where the data for each 'class' is an aggregate of several different types (for example, data from a macroclass, or as we shall see, blocks in an image), GM models are particularly valuable because they can account for local features in the data with a minimum of parameters.

We propose a number of classification methods built upon GM clustering algorithms. In Section 2, we identify three GM clustering algorithms, including two interesting algorithms (ECVQ and GMVQ) from quantization work, in addition to the more traditional EM clustering algorithm. Using these, we can generate a GM density estimate for each class from the training set of vectors. We can then classify a new vector by methods such as MAP, in which case the pdfs of the GMs are compared.

[1]This work was supported by the Stanford Undergraduate Research Program under a Minor Grant, by the National Science Foundation under NSF Grant No. CCR-0073050, and by Norsk Elektro Optikk.

We also propose an interesting method to classify whole images, which we describe more precisely in Section 3. Briefly, we break an image into smaller blocks and consider the ensemble of blocks as a sample from the mixture distribution of image blocks arising from the same image class; GM codebooks can be built for image blocks from different block-ensemble classes. To classify a test image, we match the blocks in the test image to the best class distribution. The 'codebook matching' idea has been used before, notably in speech recognition [2]. Similar work in the past with images has been concerned with classifying the blocks within one image for image segmentation purposes [1, 13], or with classifying textures that recur over the image [11], and not with directly classifying entire images that have diverse image block characteristics.

A major advantage of classifying whole images is that we avoid the time-consuming process of selecting semantic features to classify, by allowing the algorithm to automatically distinguish between classes using available information.

Following, we provide the details of our methods in Sections 2 and 3. Experiments that test these methods, their results, and a discussion follow in Sections 4, 5, and 6. Section 7 concludes the paper.

2 Gauss Mixture Density Estimation

Let ξ^n denote $\xi_1, \xi_2, ..., \xi_n$. We denote an L-component Gauss mixture by $G_{(L)} = \left\{p^L, g^L\right\}$, where p_i is the weighting or the probability of selection of the ith component so that $\sum_{i=1}^{L} p_i = 1$, and g_i is the pdf of a Gaussian random variable drawn according to $\mathcal{N}(\mathbf{m}_i, \mathbf{K}_i)$. A random variable X drawn from a Gauss mixture $G_{(L)}$ has pdf of the form $f_X(\mathbf{x}) = \sum_{i=1}^{L} p_i g_i(\mathbf{x})$, $\mathbf{x}$ being a real vector. Given sample data $\mathbf{x}^N$ (in this case, training data), we can fit a Gauss mixture distribution using the three aforementioned methods:

ECVQ The Lloyd clustering procedure [12] used in designing entropy constrained vector quantizers (ECVQ) is applied with Lagrangian formulated squared error distortion (that is, MSE distortion along with a rate term). The motivation is to use the clustering algorithm to discover local modes that can be fit with Gaussian distributions. The algorithm converges to a partition $\mathcal{P} = \{S_1, ..., S_L\}$ of the sample vectors, where S_i comprises all training vectors which are mapped into the ith codeword. To form a GM model $G_{(L)}$, a Gaussian mode is assigned to each S_i,

$$\begin{aligned} p_i &= \frac{|S_i|}{N}; \\ \mathbf{m}_i &= \frac{1}{|S_i|} \sum_{x_j \in S_i} \mathbf{x}_j; \\ \mathbf{K}_i &= \frac{1}{|S_i| - 1} \sum_{x_j \in S_i} (\mathbf{x}_j - \mathbf{m}_i)(\mathbf{x}_j - \mathbf{m}_i)^T. \end{aligned}$$

EM A popular GM clustering procedure is the expectation maximization (EM) algorithm. The goal is to maximize the expectation objective $\Pr(X^N = \mathbf{x}^N)$ over

some Gauss mixture sources from which the X_i are to be drawn i.i.d. Beginning with some GM model initialization, the following updates are made in each iteration ($G_{(L)} \to G^*_{(L)}$) to monotonically converge to a (local) maximum [1]:

$$\begin{aligned}
\nu_i(j) &= \frac{p_i g_i(\mathbf{x}_j)}{\sum_{l=1}^{L} p_l g_l(\mathbf{x}_j)}; \\
p_i^* &= \frac{1}{N}\sum_{j=1}^{N} \nu_i(j); \\
\mathbf{m}_i^* &= \frac{\sum_{j=1}^{N} \nu_i(j)\mathbf{x}_j}{\sum_{j=1}^{N} \nu_i(j)}; \\
\mathbf{K}_i^* &= \frac{\sum_{j=1}^{N} \nu_i(j)(\mathbf{x}_j - \mathbf{m}_i^*)(\mathbf{x}_j - \mathbf{m}_i^*)^T}{\sum_{j=1}^{N} \nu_i(j)}.
\end{aligned}$$

GMVQ A method used in recent work on Gauss mixture vector quantization (GMVQ) [1, 5] applies the Lloyd algorithm directly to form the Gaussian modes in a GM. This method uses a Lagrangian formulated asymmetric 'distortion' between a data point and a pdf. Define the Lagrangian distortion between $\mathbf{x}$ and a pdf f weighted by a probability p to be $\rho_\lambda(\mathbf{x}, f, p) = -\ln f(\mathbf{x}) + \lambda \ln \frac{1}{p}$. (For $\lambda = 1$, this is equivalent to a log-likelihood calculation taking into account weighting probabilities.) The Lloyd clustering algorithm then becomes a direct GM modeling algorithm. We start with a GM model initialization. During each iteration step, suppose we have a partition $\mathcal{P} = \{S_1, ..., S_L\}$ of the sample data points $\mathbf{x}^N$, associated with L Gaussian modes, then we update $G_{(L)} \to G^*_{(L)}$ as follows:

$$\begin{aligned}
p_i^* &= \frac{|S_i|}{N}; \\
\mathbf{m}_i^* &= \frac{1}{|S_i|}\sum_{x_j \in S_i} \mathbf{x}_j; \\
\mathbf{K}_i^* &= \frac{1}{|S_i|}\sum_{x_j \in S_i} (\mathbf{x}_j - \mathbf{m}_i)(\mathbf{x}_j - \mathbf{m}_i)^T; \\
S_i^* &= \left\{ \mathbf{x}_j \mid \arg\min_l \rho_\lambda(\mathbf{x}_j, p_l, g_l) = i \right\}.
\end{aligned}$$

When too few data points remain in a partition, that partition is eliminated and the points belonging to it are reassigned according to ρ_λ in the next iteration.

To avoid malformed covariance matrices (i.e. not positive definite) in Gaussian modes due to dependence or lack of sample points, we also apply a covariance regularization step at the end of the each run [8]. We write the final update as $\mathbf{K}_i^* = (1-\alpha)\mathbf{K}_i + \alpha\mathbf{M}$, for some $\alpha \in [0, 1]$ and where

$$\mathbf{M} = \frac{1}{N-L}\sum_{l=1}^{L}\sum_{x_j \in S_l} (\mathbf{x}_j - \mathbf{m}_l)(\mathbf{x}_j - \mathbf{m}_l)^T.$$

3 Whole Image Classification

In the case of image classification, we divide all images into smaller $n \times n$ blocks. The pixel values in the blocks (or some transformation of pixel values) become the vectors for classification purposes. Suppose each image is divided into B such blocks, then there are B vectors per image. During training of each class, we pool all of the vectors from the images belonging to that class. The semantic differences between classes manifest themselves in the differences in the mixture distributions of image blocks. When a test image is presented, the B vectors (or image blocks) within it allow us to estimate the source distribution of the blocks of the test image.

This suggests using distribution distance to make classification decisions, and [3, 6] provides a treatment of a distribution distance computation between Gauss mixtures. Instead we choose to implement a simpler entropy-constrained log-likelihood method to classify the collection of B vectors from the test image. Suppose that we have the GM models for the K classes of data, that is, $G_{1,(L_1)} = \left\{p_1^{L_1}, g_1^{L_1}\right\}, ..., G_{K,(L_K)} = \left\{p_K^{L_K}, g_K^{L_K}\right\}$, constructed using the GMVQ algorithm with Lagrangian distortion measure $\rho_\lambda(\mathbf{x}, p, f) = -\ln f(\mathbf{x}) + \lambda \ln \frac{1}{p}$. The image is assigned to the class that minimizes the distortion sum for the vectors $\mathbf{x}^B$ obtained from the test image. Compactly, $\mathbf{x}^B$ is assigned to the class:

$$\arg\min_k \sum_{j=1}^{B} \min_{l \in \{1,...,L_k\}} \rho_\lambda(\mathbf{x}_j, p_{k,l}, g_{k,l}).$$

4 Experiments

Our data is provided by Norsk Electro Optikk (NEO), a company that maps the interior walls of gas pipelines with an optical scanner. NEO intends to catalogue features of interest (e.g. surface characteristics) in the pipeline segments. Accurate classification of this pipeline data allows for early detection of pipeline damage, which is of significant commercial interest. The images are grayscale with size 96×128 pixels. In addition to the raw data, there is a derived dataset consisting of features (22 for each image) hand-picked for their ability to distinguish classes [9, 10].

There are, in total, 12 classes in the pipeline dataset, as described in [9], corresponding to various surface characteristics of the pipeline segments. We choose to build classifiers to distinguish three macroclasses: Plain Steel (hereafter Class S), Longitudinal Weld (Class V), and Field Joint (Class W).

Macroclass	**Component Classes**	**Sample Count**
S	Normal, Osmosis Blisters, Black Lines, Small Black Corrosion Dots, Grinder Marks, MFL Marks, Corrosion Blisters, Single Dots	153
V	Longitudinal Welds	20
W	Weld Cavity, Field Joint	39

We choose these three macroclasses because they present a realistic classification

problem to test our methods upon. The macroclasses, by their very nature, are mixtures, so GM models are well suited here.

The hand-picked (derived) dataset and the image-based dataset have very different characteristics. In the former, vector dimension is low (22) and the information is dense in the dimensions due to human effort. In the latter, vector dimension is high for the whole image ($128 \times 96 = 122880$), much of which is devoid of classifiable content. We apply the appropriate algorithm to each dataset:

- For the hand-picked features, we choose to build classifiers by modeling the source as a random variable in $\mathbb{R}^{22}$. We fit a Gauss mixture model to the training data from each macroclass separately. Final classification is by MAP. This is done for all three GM modeling methods (ECVQ, EM, and GMVQ). (We fix $\lambda = 1$ and $\alpha = 0.01$.)

- For the image-based data, we use the method described in Section 3, since practically, we cannot take the whole image as a single feature vector. Noting that the images in our dataset have been previously stored using JPEG compression and subsequently decompressed, we do two things to avoid JPEG artifacts. For each image, we divide it into 192 8×8 blocks. Instead of using raw pixel values, each 8×8 block is also Fourier transformed, and the 15 coefficients in the upper-left triangle, with the DC component at position $(1, 1)$, are taken and reshaped into a vector. (In this experiment, including higher frequency coefficients beyond the 15 does not appear to be an improvement as they contain much JPEG quantization noise.) Unrelated to JPEG compression, we take the magnitude of the Fourier transform only, discarding the phase, since we are not interested in shift variations of features in blocks. The 15 dimensional real vectors, then, are used for training with GMVQ. We train separately for the original component classes and combine the classification results into the three macroclasses as the last step. (Again we fix $\lambda = 1$ and $\alpha = 0.01$.)

For comparison, results are also obtained using other established classification methods (Regularized QDA, 1-NN, MART) [7] on the hand-picked features. MART is a gradient boosted version of a classification tree [4].[2] LDA fits a Gaussian with the same covariance to each class. QDA calculates the covariance independently for each class. Regularized QDA uses a weighted average of the LDA and QDA covariances for each class. The image is assigned to the class with highest probability. The final algorithm considered is a simple one-nearest-neighbor classifier (1-NN) using Euclidean distance.

All methods above are run on the dataset using leave-one-out cross-validation.

5 Results

The table below shows classification results from all methods described in Section 4. The first six algorithms classify hand-picked features whereas the final one classifies images using the method described in Section 3. The last four algorithms are GM based, as contrasted with the first three, which are not.

[2]MART was implemented using code available at http://www-stat.stanford.edu/~jhf/

Recall is defined to be $\frac{\#\text{ assigned correctly to class}}{\#\text{ total in class}}$, whereas precision is defined to be $\frac{\#\text{ assigned correctly to class}}{\#\text{ total assigned to class}}$. Overall accuracy, defined to be $\frac{\#\text{ correct assignments}}{\#\text{ total assignments}}$, is displayed in the rightmost column.

Method	**Recall**			**Precision**			**Accuracy**
	S	**V**	**W**	**S**	**V**	**W**	
MART	0.9608	0.9000	0.8718	0.9545	0.9000	0.8947	**0.9387**
Reg. QDA	0.9869	1.0000	0.9487	0.9869	0.9091	1.0000	**0.9811**
1-NN	0.9281	0.7000	0.8462	0.9221	0.8750	1.0000	**0.8915**
MAP-ECVQ	0.9737	0.9000	0.9437	0.9739	0.9000	0.9487	**0.9623**
MAP-EM	0.9739	0.9000	0.9487	0.9739	0.9000	0.9487	**0.9623**
MAP-GMVQ	0.9935	0.8500	0.9487	0.9682	1.0000	0.9737	**0.9717**
Image-GMVQ	0.9673	0.8000	0.9487	0.9737	0.7619	0.9487	**0.9481**

6 Discussion

On the hand-picked feature set, the GM based methods (MAP-ECVQ, MAP-EM, MAP-GMVQ) are competitive with the non-GM based methods, outperforming both 1-NN and MART. Arguably, MAP-GMVQ does equally well as regularized QDA. In fact, excepting Class V, which suffers from a paucity of training and testing data, MAP-GMVQ does somewhat better. We emphasize that we do not optimize for the best regularization coefficient α in the GM based methods, as is done in regularized QDA. We expect that in a completely equivalent comparison between MAP-GMVQ and regularized QDA, (i.e. optimizing for α in both), and with enough data, the former would do better than the latter for datasets with significant local features.

Next, we compare the three underlying GM clustering algorithms. We find that GMVQ tends to perform slightly better than EM, here and in other test cases. ECVQ, on the other hand, assumes nothing about the shape of the distribution during the clustering process, and tends to overfit the data and can perform poorly at times. Consistently accurate classification on different datasets empirically shows that GMVQ can be an excellent alternative to the more popular EM method for fitting GM models to data, considering that GMVQ converges more quickly than EM and, supplied with a Lagrangian distortion, needs no specialized pruning procedure as EM does.

Whole image classification also performs surprisingly well compared to the other methods, again outperforming MART and 1-NN. Though it is not as good as the best of the others, we must keep in mind that no class-specific features are pre-selected for this classification, which is a compelling advantage in favor of this method.

Figure 1 shows the details of this classification graphically. A large number of image blocks in images belonging to several different classes may be similar (blocks showing the background, for instance), so classes may have similar modes in their GM models. However, the image blocks that are distinctive appear as distinctive GM modes. A test image may receive similar distortions from multiple classes for those blocks characteristic of multiple classes. However, the distinctive blocks will receive a significantly lower distortion from the class to which they truly belong than from the other classes. We attribute the high performance of whole image classification in this experiment to the kind of robustness associated with examining a sample of more than

one vector during test image classification, as well as to good signal extraction in the form of the Fourier transform. Of course, other transforms, especially multiresolution transforms like wavelets, may be even more appropriate if finer control over image feature distinctions of different spatial resolutions is desired.

7 Conclusion and Future Work

We have shown empirically that Gauss mixture clustering methods developed for quantization can be adapted to a realistic classification task. Due to their good density modeling properties, GM models can provide high accuracy for classification just as well as they can provide low distortion for quantization. The GMVQ clustering algorithm appears to be an excellent alternative to the more complex EM algorithm for GM density estimation.

An area that needs further exploration in the future is the relationship between the distortion of a GM quantizer and the accuracy of a GM classifier. One aspect of the relationship is the effect of λ in the Lagrangian distortion functions. We use $\lambda = 1$ here throughout as it is a statistically meaningful value. For GMVQ, it connects distortion to log-likelihood. Other values of λ have been tried, with the obvious result of decreasing the number of Gaussian modes as λ increases; but it is still unclear what effects λ has on the final classification accuracy.

The result that most intrigues us is the good performance of whole image classification using image block ensembles. This method seems very adept at encoding locally differentiating features in the class distributions and satisfactorily classifies the dataset at hand; to a large extent, this echoes positive outcomes of similar ideas in image segmentation and image databases research [13, 14]. While we have used gas pipeline images in our experiments with encouraging results, the same approach can be applied to natural images and other images in broader contexts.

8 Acknowledgements

We would like to thank Norsk Elektro Optikk for providing the datasets used in the experiments.

References

[1] A. Aiyer, "Robust image compression using Gauss mixture models," Ph.D. Thesis, Department of Electrical Engineering: Stanford University, 2001.

[2] D.K. Burton, J.E. Shore, J. Buck, "Isolated-word speech recognition using multisection vector quantization codebooks," *IEEE Trans. Acoustics, Speech, and Signal Processing*, pp 837-49, 1985.

[3] T. Cover, J. Thomas, *Elements of Information Theory*, John Wiley & Sons, New York, 1991.

[4] J. Friedman, "Greedy function approximation: A gradient boosting machine," *The Annals of Statistics*, vol. 39, no. 5, 2001.

[5] R.M. Gray, T. Linder, "Mismatch in high rate entropy constrained vector quantization," Vol. 49, pp. 1204–1217g, *IEEE Trans. Inform. Theory*, May, 2003.

[6] R.M. Gray, J. Young, A. Aiyer, "Minimum discrimination information clustering: modeling and quantization with gauss mixtures," *Proceedings 2001 IEEE ICIP*, Thessaloniki, Greece, 2001.

[7] T. Hastie, R. Tibshirani, J. Friedman, *The Elements of Statistical Learning*, Springer-Verlag, New York, 2001.

[8] J.P. Hoffbeck, D.A. Landgrebe, "Covariance Matrix Estimation and Classification with Limited Training Data," *IEEE Trans. Pattern Analysis and Machine Intelligence*, pp 763-7, 1996.

[9] D.B. O'Brien, M. Gupta, R.M. Gray, J.K. Hagene, "Automatic Classification of Images from Internal Optical Insepection of Gas Pipelines," *ICPIIT VIII Conference 2003*, Houston.

[10] D.B. O'Brien, M. Gupta, R. M. Gray, J. K. Hagene, "Analysis and classification of internal pipeline images," *Proceedings of ICIP 2003*, Barcelona, Spain.

[11] K. Pyun, C.S. Won, J. Lim, R.M. Gray, "Texture classification based on multiple Gauss mixture vector quantizers," *Multimedia and Expo, 2002*, pp 501-4, 2002.

[12] J. Shih, A.K. Aiyer, R.M Gray, "A Lagrangian formulation of high rate quantization," *Proceedings 2001 IEEE ICASSP*, pp 2629-32, Salt Lake City, 2001.

[13] S. Yoon, K. Pyun, C.S. Won, R.M. Gray, "Image classification using GMM with context information and reducing dimension for singular covariance," *DCC 2003.*

[14] C. Young, "Clustered Gauss mixture models for image retrieval," Ph.D. Thesis, Department of Electrical Engineering: Stanford University, 2003.

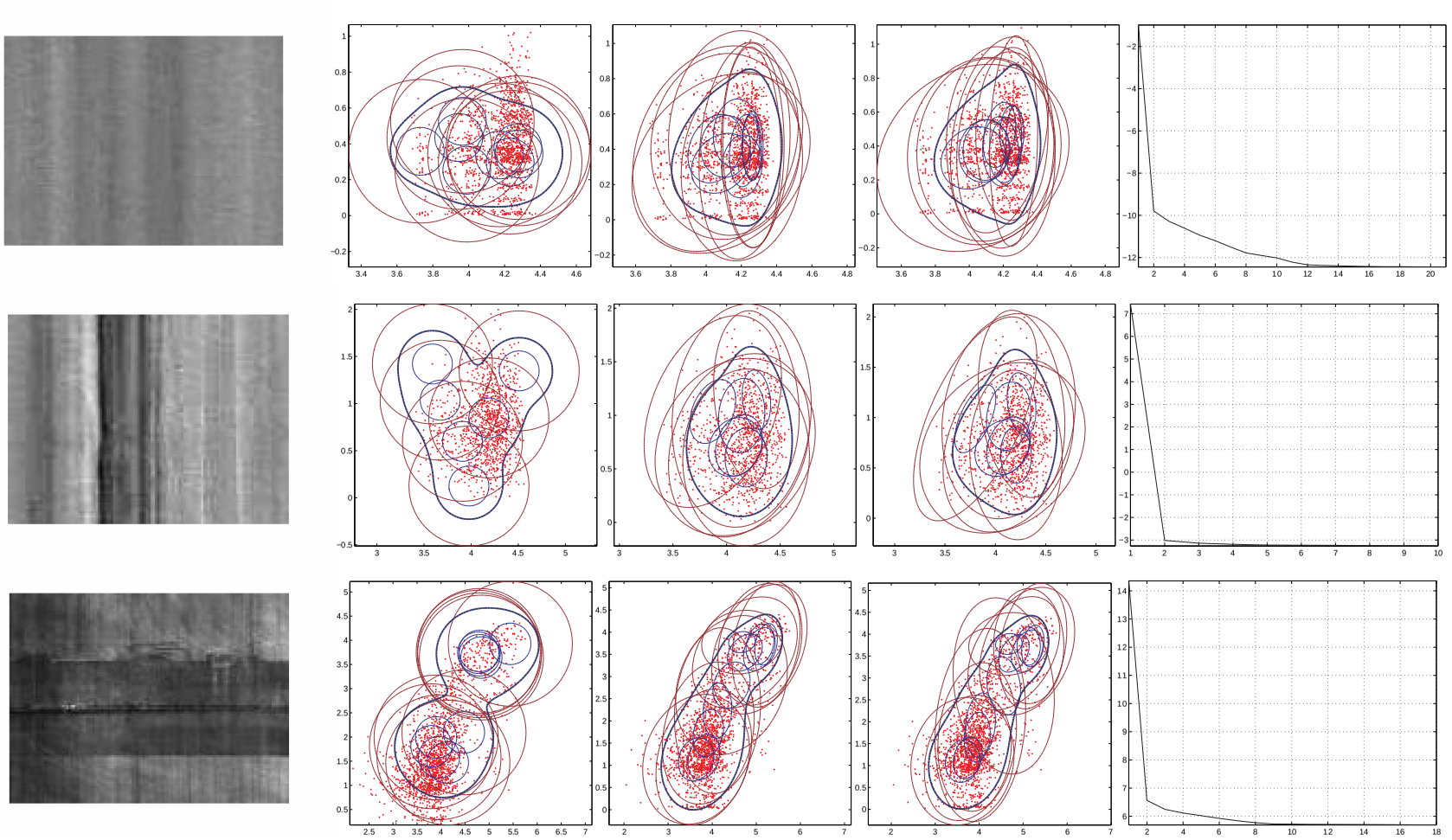

Figure 1: On the left are example images from Class S, Class V, and Class W (top to bottom). On each corresponding row, the GMVQ algorithm is shown converging to a solution: The middle three plots show (left to right) the random initialization, the solution after one iteration, and the converged solution. The dots are high-dimensional training vectors (15 low frequency coefficients in the FFT of 8-by-8 image blocks) projected onto the first two dimensions; the thin curves are the first and third standard deviations of each Gaussian mode; the thick curves are level sets of Gauss mixture pdfs. Rightmost are plots of the Lagrangian distortions vs. iteration count.