A Vector Quantization Schema for Non-Stationary Signal Distributions Based on ML Estimation of Mixture Densities

N. A. Vlassis K. Blekas G. Papakonstantinou A. Stafylopatis

ABSTRACT

We show that by selecting an appropriate distortion measure for the encoding-decoding vector quantization schema of signals following an unknown probability density $p(x)$, the process of minimizing the average distortion error over the training set is equivalent to the Maximum Likelihood (ML) estimation of the parameters of a Gaussian mixture model that approximates $p(x)$. Non-stationary signal distributions can be handled by appropriately altering the parameters of the mixture kernels.

1 Introduction

Vector quantization (VQ) [1] is a data compression method in signal processing in which an input signal x is assigned a value $c(x)$ by an encoder, and this value—instead of the actual value x—is sent through a communication channel to the receiver. The latter applies a decoder function $x' = x'(c(x))$ to obtain the original value. The quality of this quantizer is measured by the average distortion D over a training set $\mathcal{T} = \{x_1, \ldots, x_n\}$ of d-dimensional input signals defined as

$$D = \sum_{i=1}^{n} \delta(x_i, x_i'), \qquad (1)$$

with $\delta(x, x')$ being the distortion, i.e., a dissimilarity measure, between the actual signal x and its reconstruction x'. The optimal values for the encoder and the decoder of a VQ schema are those that minimize the function D.

We assume that the probability density function $p(x)$ of the incoming signals is unknown and we approximate it with a general mixture of K Gaussian kernels [4, 8], each one parametrized on its mean μ_j and variance σ_j^2. These parameters are to be estimated by the Maximum Likelihood (ML) method [5] over the training set $\mathcal{T}$. In the following we show that the ML estimation is equivalent to minimizing the average distortion function D, thus provides a way to compute the nearly optimal values for the encoder and decoder functions. In addition, we propose a VQ schema that can handle non-stationary signal distributions.

A similar model has been proposed in the literature under the name of Probabilistic Neural Networks [7], in which the mixing weights are assumed equal among all kernels and equal to the reciprocal of the total number of input samples. For the estimation of the total number of kernels the reader may refer to [8] for mathematical methods, or [6] for a neural network approach.

2 The Vector Quantizer

We assume that the total input density $p(x)$ is a mixture of K Gaussian kernels

$$p(x) = \sum_{j=1}^{K} \pi_j f_j(x), \qquad (2)$$

where each kernel j is the normal probability density function

$$f_j(x) = \frac{1}{\sigma_j \sqrt{2\pi}} \exp[\frac{-(x-\mu_j)^2}{2\sigma_j^2}], \qquad (3)$$

parametrized over its mean μ_j and variance σ_j^2, and having prior probability π_j. Additionally, it must hold $\sum_{j=1}^{K} \pi_j = 1$, $\pi_j \geq 0$.

The posterior probability that a new signal x is assigned to kernel k is given by the Bayes

formula

$$P\{k|x\} = \frac{\pi_k f_k(x)}{\sum_{j=1}^{K} \pi_j f_j(x)}, \quad (4)$$

and the Bayes minimum risk rule assigns a signal x to the kernel k with the maximum posterior probability $P\{k|x\}$. Based on the above, we define the VQ encoding function $c(x)$ to be the minimum risk kernel, i.e.,

$$c(x) = k \quad \text{if} \quad P\{k|x\} = \max P\{j|x\}, \quad (5)$$

for $j = 1, \ldots, K$, and the decoding function $x'(c(x))$ as

$$x'(c(x)) = \mu_{c(x)}, \quad (6)$$

i.e., the mean of the kernel the input signal was assigned to.

The Maximum Likelihood procedure for the training set $\mathcal{T}$ assigns to the parameters π_j, μ_j, and σ_j of each kernel j values that maximize the log-likelihood

$$L(\mathcal{T}) = \sum_{i=1}^{n} \ln p(x_i) \quad (7)$$

which, by applying the logarithm to (3) and using (2) and (4), amounts to minimizing the total distortion D of (1), with $x'_i = x'(c(x_i))$ and

$$\delta(x_i, x'_i) = \frac{(x_i - x'_i)^2}{\sigma^2_{c(x_i)}} + 2\ln\sigma_{c(x_i)} - 2\ln\pi_{c(x_i)} + 2\ln P\{c(x_i)|x_i\}, \quad (8)$$

with $c(x_i)$ defined from (5) while the first term $(x_i - x'_i)^2/\sigma^2_{c(x_i)}$ defines the Mahalanobis distance from x_i to the mean $x'_i = x'_i(c(x_i))$ of the kernel $c(x_i)$. Eq. (8) defines the distortion measure of our VQ schema.

3 Estimating the unknown density

Maximizing the log-likelihood of (7) with respect to π_j, μ_j, and σ_j^2, it can be shown [9, 10] that recursive expressions for the estimation of the parameters of each Gaussian kernel j can be estimated as

$$\mu_j := \mu_j + (n\pi_j)^{-1} P\{j|x\}(x - \mu_j), \quad (9)$$

$$\sigma_j^2 := \sigma_j^2 + (n\pi_j)^{-1} P\{j|x\}[(x-\mu_j)^2 - \sigma_j^2], \quad (10)$$

$$\pi_j := \pi_j + n^{-1}(P\{j|x\} - \pi_j), \quad (11)$$

applied each time a new input signal x is arrived, while $P\{j|x\}$ is the posterior probability (4) that x is assigned to kernel j.

Substituting the number n—the cardinality of $\mathcal{T}$—above with a constant l, i.e., rendering the system 'memoriless' to old signals, non-stationary signal distributions can also be handled [10]. Also, we propose here a method that on-line seeks for the correct number of kernels based on simple test statistics for testing the hypothesis of single normality against a two-kernel alternative.

3.1 Testing for the number of kernels

Split: We first look for a test statistic to decide when a kernel should split in two. A statistical test is needed to check the hypothesis that the input samples assigned to a particular kernel with μ and σ follow a single Gaussian against the alternative that they follow a mixture of two kernels, in which case the single kernel should split in two.

We form a simple sequential test statistic based on a weighted formula of the kurtosis, or fourth moment, of a kernel j as

$$k_j := k_j + (n\pi_j)^{-1} P\{j|x\} \left[\left(\frac{x - \mu_j}{\sigma_j} \right)^4 - k_j - 3 \right], \quad (12)$$

with μ_j and σ_j the current ML estimates for the parameters of the kernel. On the hypothesis that x_i follow a normal distribution $N(\mu_j, \sigma_j)$ it follows that the random variable

$$q = k_j \sqrt{n\pi_j / 96} \quad (13)$$

approximately follows normal distribution $N(0, 1)$, and thus we can accept the hypothesis that the kernel j is $N(\mu_j, \sigma_j)$ if q is sufficiently close to zero.

When a new kernel is created it gets a zero value of kurtosis which is updated at each iteration step from (12). The first time the aforementionted kurtosis test is violated we split the kernel and create two kernels with means $\mu + \sigma$ and $\mu - \sigma$, and variances and priors both equal to the original variance. The priors of all kernels are then re-normalized to ensure $\sum_{j=1}^{K} \pi_j = 1$.

Join: A reasonable criterion for joining two neighboring kernels in one is when they have almost the same variance and very near

means. For checking for equality of the variances σ_j^2 and σ_k^2 of two neighboring kernels j and k we form the ratio of the larger to the smaller variance, e.g.,

$$F = \frac{\sigma_j^2}{\sigma_k^2}, \tag{14}$$

and accept the hypothesis of equal variances if F is lower than a pre-determined threshold.

If the test for equal variances succeeds, we subsequently check for equal means assuming common variance $\sigma^2 = \sigma_j \sigma_k$, using the test statistic

$$t = \sqrt{n\pi_j}\, \frac{\mu_j - \mu_k}{\sigma\sqrt{2}},$$

which under the hypothesis of equal means approximately follows normal distribution $N(0,1)$. Similarly, we accept the hypothesis of equal means if t is sufficiently close to zero. Then the two kernels are joined in one with mean $(\mu_j + \mu_k)/2$, variance σ^2, and prior equal to π_j. The priors of all kernels are renormalized to unity.

Removing a kernel: A kernel j is removed from the mixture when its prior probability π_j is below $1/n$, a threshold ensuring that the terms in (9) and (10) remain bounded. After a kernel is removed all kernels should update their priors to sum one.

4 Multivariate densities

For problems of higher dimension d, Eq. (3) generalizes for a kernel j and input vector $\mathbf{x} = [x_1, \ldots, x_d]$ to

$$f_j(\mathbf{x}) = \frac{\exp[-0.5(\mathbf{x} - \mathbf{m}_j)\mathbf{S}_j^{-1}(\mathbf{x} - \mathbf{m}_j)^T]}{\sqrt{(2\pi)^d \det \mathbf{S}_j}} \tag{15}$$

where $\mathbf{m}_j = [\mu_{j1}, \ldots, \mu_{jd}]$ is the mean of the kernel, $\mathbf{S}_j$ is the covariance matrix, and $\det \mathbf{S}_j$ denotes the determinant of $\mathbf{S}_j$.

The approach we described in the previous section can be directly applied to the multivariate case if we make the assumption that in each multivariate kernel j the d components of the input vector $\mathbf{x}$ are jointly normal and uncorrelated, an assumption that results in *hyper-ellipsoidal* kernels. In this case [3] the respective covariance matrix $\mathbf{S}_j$ is diagonal and (15) can be written as the product of the d marginal univariate Gaussians, i.e.,

$$f_j(\mathbf{x}) = \frac{1}{\sigma_{j1} \cdots \sigma_{jd}\sqrt{(2\pi)^d}} \exp[\frac{-(x_1 - \mu_{j1})^2}{2\sigma_{j1}^2} + \cdots + \frac{-(x_d - \mu_{jd})^2}{2\sigma_{jd}^2}] \tag{16}$$

where $\sigma_{j1}^2, \ldots, \sigma_{jd}^2$ are the diagonal elements of $\mathbf{S}_j$. For the prior updates we use (2), (4), and (11) with the kernel densities substituted from (16), while adaptation of the kernel parameters is done in each dimension separately so that the d components of $\mathbf{m}_j$ and the d diagonal elements of $\mathbf{S}_j$ of each kernel are estimated as in the univariate case from (9) and (10), respectively. The kurtosis test is applied on each dimension separately and if it succeeds for dimension i then the split kernels keep all but the i components unaltered, the latter being changed as in the univariate case. Finally, to join two kernels, the respective tests must succeed in all dimensions.

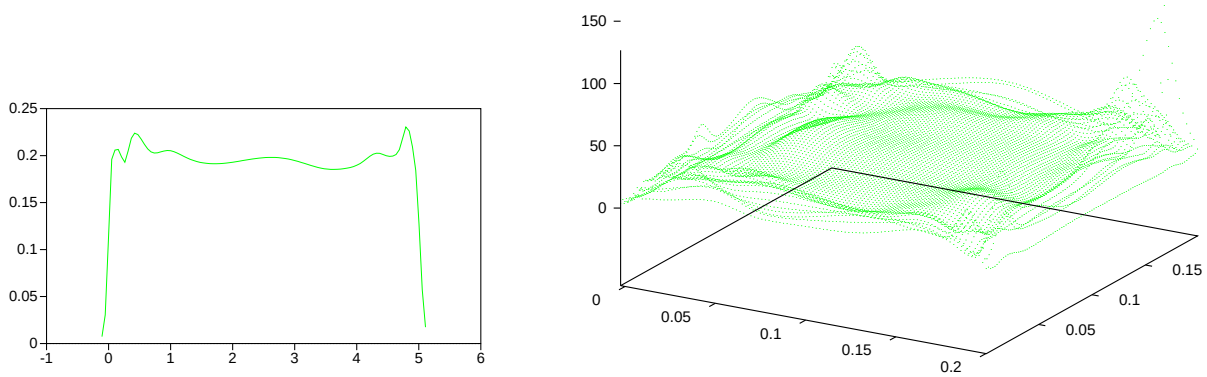
5 Examples

To test the validity of our method we applied it to the problem of estimating a uniform distribution, in 1-d and 2-d. This can be considered a difficult problem for Gaussian approximators, and although theoretically [5] a universal approximator for continuous functions via Gaussians can be established, in practice this proves to be a hard task.

In Fig. 1 we show the approximation of our algorithm to two uniform densities, the first 1-d and the second 2-d. One notes that the functions are adequately approximated near the center, with the expected "jumps" at the discontinuities (corners). For the first, the number of employed kernels was 31, while for the second 170. The thresholds for the statistical tests for splitting, joining, and removing a kernel, were kept to a minimum to favor the creation of many kernels and thus a better approximation.

6 Conclusions-Discussion

We showed that the problem of minimizing the average distortion (1) in a VQ schema is equivalent to the ML estimation procedure for the unknown signal density. We also described an iterative self-organizing procedure for estimating the unknown probability density function of the input signals. The bulk



a. 1-d uniform (K=31)

b. 2-d uniform (K=170)

Figure 1: Estimating a uniform density in a 1-d problem (a), and a 2-d problem (b). Parameters: $n = 1000$, $a = 0.1$.

of our approach lies in approximating the unknown density with a mixture of Gaussian kernel functions and employing the Maximum Likelihood technique for estimating the parameters of each kernel. Moreover, by appropriately splitting and joining kernels it is possible to handle even non-stationary signal distributions.

We may note the similarity of our kernel means update equation (9) to the reconstruction vectors update formula of [2]

$$x'(c'(x)) = x'(c'(x)) + \eta\pi(c'(x) - c(x))(x - x'(c'(x))). \quad (17)$$

There η is the learning rate and $\pi(\cdot)$ is the pdf of a noise added by the channel to the code $c(x)$ to produce a distorted code $c'(x)$ at the receiver. From the above relationship we infer a similar model for the channel noise in our VQ schema; the noise pdf over a codevector $c(x)$ is the Gaussian kernel pdf $p_{c(x)}$.

References

[1] A. Gersho and R. M. Gray. *Vector Quantization and Signal Compression*. Kluwer Academic Publishers, Boston, 1992.

[2] S. P. Luttrell. Self-organization: A derivation from first principle of a class of learning algorithms. In *Proc. IEEE Conf. on Neural Networks*, pages 495–498, Washington, DC, 1989.

[3] A. Papoulis. *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, 3rd edition, 1991.

[4] R. Redner and H. Walker. Mixture densities, maximum likelihood and the EM algorithm. *SIAM Review*, 26(2):195–239, Apr. 1984.

[5] B. D. Ripley. *Pattern Recognition and Neural Networks*. Cambridge University Press, Cambridge, U.K., 1996.

[6] S. Shimoji. *Self-Organizing Neural Networks Based on Gaussian Mixture Model for PDF Estimation and Pattern Classification*. PhD thesis, University of Southern California, 1994.

[7] D. Specht. Probabilistic neural networks. *Neural Networks*, 3:109–118, 1990.

[8] D. Titterington, A. Smith, and U. Makov. *Statistical Analysis of Finite Mixture Distributions*. Wiley, 1985.

[9] H. G. C. Tråvén. A neural network approach to statistical pattern classification by "semiparametric" estimation of probability density functions. *IEEE Transactions on Neural Networks*, 2(3):366–377, May 1991.

[10] N. A. Vlassis, A. Dimopoulos, and G. Papakonstantinou. The probabilistic growing cell structures algorithm. In *Proc. ICANN'97, 7th Int. Conf. on Artificial Neural Networks*, pages 649–654, Lausanne, Switzerland, Oct. 1997.