ENTROPY AND MEMORY CONSTRAINED VECTOR QUANTIZATION WITH SEPARABILITY BASED FEATURE SELECTION

Sangho Yoon and Robert M. Gray

Information Systems Lab., Department of Electrical Engineering, Stanford University
{holyoon,rmgray}@stanford.edu

ABSTRACT

An iterative model selection algorithm is proposed. The algorithm seeks relevant features and an optimal number of codewords (or codebook size) as part of the optimization. We use a well-known separability measure to perform feature selection, and we use a Lagrangian with entropy and codebook size constraints to find the optimal number of codewords. We add two model selection steps to the quantization process: one for feature selection and the other for choosing the number of clusters. Once relevant and irrelevant features are identified, we also estimate the probability density function of irrelevant features instead of discarding them. This can avoid the bias of problem of the separability measure favoring high dimensional spaces.

1. INTRODUCTION

In vector quantization (VQ) [5] an input vector is represented by one of a predefined set of patterns on the basis of which pattern is closest to the given input vector. The encoder and decoder in VQ are associated with partitions (clusters) and codewords (cluster centers), respectively. Thus VQ design can be viewed as a clustering algorithm. In addition VQ can be viewed as fitting a model when partition cells are represented by their conditional probability density functions (pdf) and prior probabilities are weights. In particular, we are interested in fitting Gauss mixture models (GMM) to data using Gauss mixture VQ (GMVQ) [7] because GMM has been used successfully in various areas in signal processing and shown to be robust [3]. The most popular approach to fitting a GMM to data is the EM algorithm [17], but the Lloyd algorithm [9][7] provides an alternative. The Lloyd algorithm is an iterative algorithm: it first assigns data to the closest cluster centers, next updates cluster centers, and then iterate these two steps until convergence is reached. The EM algorithm makes soft decisions for input data, whereas the Lloyd algorithm makes hard decisions. See [7] for more details.

In pattern recognition it is important to find informative (or relevant) features and the underlying distribution of given data. This can be also applied to VQ. We need to decide which features to use in VQ design. We will consider only feature selection in this paper for its simplicity and the interpretability of the effect of features on learning. Many algorithms have been suggested for feature selection [4]. We largely follow a well-known feature selection criterion of [14] that tries to maximize separability among clusters in selecting features. Since clusters are more separable in higher dimensional spaces, the separability based criterion can prefer higher dimensional spaces without necessarily finding relevant features. We try to avoid this bias problem by keeping irrelevant features and estimating their distribution instead of discarding them.

If we use GMM to represent the underlying distribution of data (or fit GMM to data), then we need to estimate mean vectors, covariance matrices, prior probabilities and the number of Gaussian components in GMM. Here the number of Gaussian components corresponds to the number of clusters. Parameters except the number of Gaussian components can be estimated by a sample based approach with (or without) regularization [6][8]. Estimating the number of clusters has been actively studied over the years and many algorithms have been suggested, including an EM based approach with complexity penalties [19], a Bayesian approach [13], and an information theoretic approach [12]. See also [4][18]. This is not only an important problem in unsupervised learning where we do not know how many classes exist, but also is an important issue in supervised learning when we need to decide the number of components to fit a mixture model to each class.

A more important fact is that feature selection and estimating the number of clusters can not be separated. They need to be optimized jointly. We consider them as parameters to be optimized in VQ design and embody optimization steps for model selection into the classic Lloyd algorithm. We try to optimize them iteratively one at a time.

2. BACKGROUND

A vector quantizer of dimension p and size N is made up with an *encoder* α, a *decoder* β and a *length function* l. An encoder α is a mapping of an input vector x in p-dimensional Euclidean space, $\mathcal{R}^p$ into an index $i \in \mathcal{I} = \{1, 2, ..., N\}$, and

This work was partially supported by the National Science Foundation under NSF Grants CCR-0903701.

α is associated with partition $S = \{S_i, i = 1, 2, ..., N\}$ such that $S_i = \{x : \alpha(x) = i\}$. A decoder β converts the index into a source reproduction $\widehat{x}$, and β is associated with a reproduction codebook $\mathcal{C} = \{\beta(i) : i \in \mathcal{I}\}$. Finally a *length function* $\{l(i) : i \in \mathcal{I}\}$ is *admissible* if $\sum_{i\in\mathcal{I}} e^{-l(i)} \leq 1$. For a *fixed-rate* quantizer, $l(i)$ is fixed at $\ln(N)$ for all i. Otherwise a quantizer is said to be *variable-rate*. We denote a quantizer q as $q(x) = \beta(\alpha(x))$.

The performance of a quantizer is measured by average distortion between input X and its reproduction $\widehat{X} = \beta(\alpha(X))$, and rate. If X has a pdf f, average distortion and rate are defined as $E_f(q) = E_f d(X, \beta(\alpha(X))$ and $R_f(q) = E_f l(\alpha(X))$, respectively when E_f denotes expectation with respect to f. A Lagrangian combination of average distortion and rate is used to find the optimal q in the entropy-constrained VQ (ECVQ) [1]. By using a Lagrangian multiplier $\lambda > 0$, we define the Lagrangian distortion $\rho(\lambda, x, i) = d(x, \beta(i)) + \lambda l(i)$, and the expected Lagrangian distortion is

$$\rho(\lambda, f, q) \quad = \quad E_f(d(X, \beta(\alpha(X))) + \lambda l(\alpha(X))) \quad (1)$$

Then Lloyd's necessary conditions for a variable-rate quantizer q to be optimal are

- For a given decoder β, and length function l, the optimal encoder $\alpha(x) = \text{argmin}_i(d(x, \beta(i)) + \lambda l(i))$.
- For a given encoder α, and length function l, the optimal decoder $\beta(i) = \text{argmin}_y E(d(X, y|\alpha(X)))$ if the minimum exists.
- For a given encoder α, and decoder β, the optimal length function $l(i) = \text{-ln}(Pr(\alpha(X) = i))$.
- Pr$(\alpha(X) = i) \neq 0$ for $i \in \mathcal{I}$.

For a fixed rate quantizer q, $l(i)$ is fixed and rate does not play a role in codebook design. This is equivalent to $\lambda = 0$ and corresponds to the generalized Lloyd algorithm [5].

VQ can be thought of as a classifier in the sense that they both assign input data to the closest partitions or clusters. A classifier assigns each observed sample to one of a collection of clusters based on a discriminant rule. A discriminant rule is usually defined to minimize misclassification risk [6], which is the expected cost or loss ($L(i, k)$, $1 \leq i, k \leq N$) of classifying a sample to cluster i when it actually belongs to cluster k. By using a simple 0-1 loss function [6], the risk function in [6] reduces to choosing the k that maximizes $f_k(X)p_k$, where p_k and $f_k(X)$ are the prior probability of cluster k and the cluster conditional pdf of cluster k, respectively. If we model $f(X)$ by a GMM and take the negative log of $f_k(X)p_k$, the risk function in [6] becomes the following:

$$i \quad = \quad \text{argmin}_{1\leq k\leq N} d(x, \mu_k, \Sigma_k) - \log(p_k) \quad (2)$$

where $d(x, \mu_k, \Sigma_k) = \frac{1}{2}(x-\mu_k)^t\Sigma_k^{-1}(x-\mu_k) + \frac{1}{2}\log((2\pi)^p|\Sigma_k|)$, and μ_k and Σ_k are mean vector and covariance matrix of cluster k. (2) is equivalent to the first necessary condition of the Lloyd algorithm when $\lambda = 1$.

3. MODEL SELECTION

3.1. Feature Selection based on Separability Measure

We use a separability based criterion to select features. Separability can be measured by the within class scatter matrix S_w and the between class scatter matrix S_b, and they are defined as follows:

$$S_w = \sum_{k=1}^{N} p_k\Sigma_k \quad (3)$$

$$S_b = \sum_{k=1}^{N} p_k(\mu_k - \mu)(\mu_k - \mu)^t \quad (4)$$

$$\Sigma_k = E\left((X - \mu_k)(X - \mu_k)^t|\alpha(X) = k\right) \quad (5)$$

$$\mu_k = E(X|\alpha(X) = k) \quad (6)$$

$$\mu = E(X) = \sum_{k=1}^{N} p_k\mu_k \quad (7)$$

where μ_k and Σ_k are mean vector and covariance matrix of cluster k, and p_k is the prior probability of cluster k.

S_w and S_b measure within cluster scatter and between cluster scatter, respectively. We follow $trace(S_w^{-1}S_b)$ criterion [14] to select features. The larger $trace(S_w^{-1}S_b)$ is, the more separable clusters are. $trace(S_w^{-1}S_b)$ is invariant to any nonsingular linear transformation, but it prefers higher dimensional spaces: trace$(S_w^{-1}S_b)$ monotonically increases with respect to dimension when there is no change in the clustering assignments [14]. We try to ameliorate this bias problem by keeping and estimating irrelevant features instead of discarding them.

3.2. Optimizing Codebook Size

We follow a recently proposed algorithm by Gray and Gill [2] to optimize the codebook size. Using a Lagrangian formulation combining both entropy and log codebook size:

$$\rho(f, \lambda, \eta, q) = E_f\left(d(X, \hat{X}) + \lambda[(1 - \eta)l(\alpha(X)) + \eta\ln N]\right) \quad (8)$$

where $\hat{X} = \beta(\alpha(X))$, N is the codebook size, $\lambda > 0$ and $\eta \in [0, 1]$.

4. VECTOR QUANTIZATION WITH MODEL SELECTION

In the classic Lloyd algorithm, the encoder, decoder and length function are optimized in turn for each given that the other two are fixed. In our approach, we view the model selection problem as an optimization problem and embody it into our process of optimizing the codebook. More specifically, our algorithm incorporates model selection into entropy-constrained vector quantization [1].

Applying the distortion model in (2) to the Lagrangian (8), we have the following Lagrangian

$$\rho(f,\lambda,\eta,q,p) = E_f\left(d(X,\mu_{\alpha(X)},\Sigma_{\alpha(X)}) + \lambda((1-\eta)\log(p_{\alpha(X)}) + \eta \ln N)\right) \quad (9)$$

where $d(X,\mu_{\alpha(X)},\Sigma_{\alpha(X)}) = \frac{1}{2}(X-\mu_{\alpha(X)})^t\Sigma^{-1}_{\alpha(X)}(X-\mu_{\alpha(X)})$ + $\frac{1}{2}\log\left((2\pi)^p|\Sigma_{\alpha(X)}|\right)$, $X \in \mathcal{R}^p$, and the codebook size N.

To find relevant features, we assume that a feature vector X can be partitioned into two subvectors: a relevant subvector X_r and an irrelevant subvector X_{ir}.

$$X = (X_r, X_{ir}) \quad (10)$$

where $X \in \mathcal{R}^p$, $X_r \in \mathcal{R}^m$, $X_{ir} \in \mathcal{R}^{p-m}$, and $1 \le m \le p$.

We try to find optimal X_r that maximizes

$$Sep(m,N) = trace(S_w(m,N)^{-1}S_b(m,N)) \quad (11)$$

where $S_w(m,N)$ and $S_b(m,N)$ are now

$$\begin{aligned} S_w(m,N) &= \sum_{k=1}^{N} p_k\Sigma_{k,m} \\ S_b(m,N) &= \sum_{k=1}^{N} p_k(\mu_{k,m}-\mu)(\mu_{k,m}-\mu)^t \\ \Sigma_{k,m} &= E\left((X-\mu_{k,m})(X-\mu_{k,m})^t|\alpha(X_r)=k\right) \\ \mu_{k,m} &= E(X|\alpha(X_r)=k) \\ \mu &= E(X) \end{aligned}$$

We add the following two optimality conditions to the Lloyd algorithm in Section 2 using (8) and (9):

- For a given encoder α, decoder β, length function l, and dimension of feature vector p, the optimal codebook size N is the size of codebook $\mathcal{C}$ such that there is no codebook $\mathcal{C}' \subset \mathcal{C}$ for which

$$\rho(f',\lambda,\eta,q',p) < \rho(f,\lambda,\eta,q,p), \quad (12)$$

  where f' and q' are obtained from f and q by selecting $|\mathcal{C}'|$ codewords of $\mathcal{C}$.

- For a given encoder α, decoder β, length function l, and codebook size $N = |\mathcal{C}|$, the optimal dimension of relevant feature X_r is m such that there is no $m' < m$ for which

$$Sep(m,N) \le Sep(m',N') \quad (13)$$

  where $N' \le N$.

As the Lloyd algorithm iterates, we can improve a codebook $\mathcal{C}$ by removing one or more codewords as long as the increase in $E_f(d(X,\beta(\alpha(X)))$ + $\lambda[(1-\eta)l(\alpha(X))$ is less than the decrease in $\lambda\eta \ln N$, and we can improve a relevant feature vector X_r by moving one or more features of X_r to X_{ir} as long as (13) does not decrease.

5. EXPERIMENTAL RESULTS

We test our algorithm on a synthetic data set and two real word data sets from the UCI learning repository. We compare our algorithm with algorithms by Dy et al [15]. They considered both EM and *k*-means (fixed-rate VQ using MSE) to optimize choosing the number of clusters and selecting features with a vast amount of experiments. They considered two feature selection criteria: a scatter matrix criterion (termed TR hereafter; refer to [15] for more details) and a maximum likelihood (ML) criterion, and a penalty term is added to the log-likelihood to find the number of clusters. To reduce the complexity of searching in feature selection they used sequential forward search, but they performed exhaustive searching to find the number of clusters. We follow their evaluation measures: ten-fold cross validation error, recall, and precision.

In both synthetic and real world data sets, we know the class memberships of data, but we only use this information to measure the performance of algorithms in comparison. Thus we first perform unsupervised learning and let our algorithm find clusters with choosing the number of clusters and relevant features by itself. Then we label each cluster by the majority of feature vectors assigned to it. In testing, we assign each data point to the one with the smallest Lagrangian distortion and classify it by the label of the closest cluster. We measure the cross-validation error by averaging the misclassification error ratio. Recall is defined as the number of relevant features in the selected subset divided by the total number of relevant features, and precision is defined as the number of relevant features in the selected subset divided by the total number of features selected. Recall and precision are used to measure the algorithm's ability to select relevant features.

We fit GMM to data by using the Lloyd algorithm with two model selection steps. We set $\lambda = 1/(1-\eta)$ to reduce the number of parameters in (9), and we find that $\eta \in [0.7, 0.85]$ gives us the best results in most cases. An initial codebook is obtained by the splitting algorithm [2].

Our synthetic data set has four equiprobable Gaussian clusters with means at (0,0),(1,4),(5,5) and (5,0), and covariances equal to I. We add three Gaussian normal random noise features. Table 1 shows that in all aspects our Lloyd algorithm with pruning steps is superior to EM based model selection algorithms and finds the number of clusters and relevant features perfectly.

Table 2 shows comparisons of ten-fold cross validation on the iris and wine data sets. The iris data set has three classes, four features and 150 samples. The wine data set has three classes, thirteen features and 178 samples. In both data sets, and our algorithm shows the minimum CV error and finds the number of clusters perfectly trying to find relevant features. In the wine data set, *k*-means shows the best performance in terms of CV error, but it uses all features with the number of clusters given. Thus, in both data sets, our algorithm is superior to any other model selection algorithms compared.

Table 1. Ten-fold cross validation results on a synthetic data. The data set has 500 samples. CV error represents the average ten-fold cross-validation misclassification error. Numbers in the parenthesis are standard deviations. FSSEM-k-ML and FSSEM-k-TR stand for feature selection algorithms by ML and Trace criteria with searching for the number of codewords based on EM by Dy et al. EM-k represents EM algorithm with finding the number of clusters and fixing feature set.

Four-cluster data set

	Lloyd	FSSEM-k-ML	FSSEM-k-TR	EM-k
CV error(in %)	**3.4(1.3)**	4.0(2.2)	4.0(2.0)	48(9.5)
Avg # of clusters	**4.0(0)**	4.0(0)	4.0(0)	2.0(0)
Avg precision	**0.9(0.16)**	0.5(0)	0.53(0.07)	0.2(0)
Avg recall	**1.0(0)**	1.0(0)	1.0(0)	1.0(0)

Table 2. Ten-fold cross validation results on the real world data sets. FSS-Kmeans-k-ML and FSS-Kmeans-k-TR stand for feature selection algorithms by ML and Trace criteria with searching for the number of codewords based on *k*-means by Dy et al.

Iris data set

	% CV error	Avg # of clusters	Avg # of features
Lloyd	**0.67(2.1)**	**3(0)**	**2.8(0.4)**
FSSEM-k-ML	3.3(4.5)	3.1(0.3)	2.7(0.5)
FSSEM-k-TR	4.7(5.2)	3.0(0)	2.5(0.5)
FSS-Kmeans-k-ML	4.7(4.3)	3.4(0.5)	2.4(0.5)
FSS-Kmeans-k-TR	13.3(9.4)	4.5(0.7)	2.3(0.5)
EM	3.3(5.4)	fixed at 3	fixed at 4
k-means	16.7(4.5)	fixed at 3	fixed at 4

Wine data set

	% CV error	Avg # of clusters	Avg # of features
Lloyd	**4.1(4.8)**	**3(0)**	**5.1(1.2)**
FSSEM-k-ML	21.2(10.9)	3.9(0.8)	3.2(0.9)
FSSEM-k-TR	12.4(13.0)	3.6(0.8)	3.8(1.8)
FSS-Kmeans-k-ML	16.1(7.1)	4.1(0.3)	3.4(1.0)
FSS-Kmeans-k-TR	22.8(11.1)	3.4(0.5)	2.7(1.3)
EM	10.0(17.3)	fixed at 3	fixed at 13
k-means	1.2(2.4)	fixed at 3	fixed at 13

6. CONCLUSIONS

Our algorithm extends the Lloyd algorithm to seek the optimum number of clusters and relevant features in an iterative way without discarding irrelevant features. Contrary to the conventional model selection algorithms where optimization is performed for each model and the best one is chosen by comparing all models with complexity penalties, our proposed algorithm selects the optimal model as we optimize our codebook design. We add two necessary conditions for model selection to the classic Lloyd algorithm. This reduces the complexity of model selection significantly. Experimental results on both synthetic and real world data sets show that our algorithm is superior to the state of the art of model selection algorithms on these data sets.

7. REFERENCES

[1] P.A. Chou, T. Lookabaugh, and R.M. Gray, "Entropy-constrained vector quantization," *IEEE Trans. on ASSP,* Vol. 37, No. 1, pp.31-42, January 1989.

[2] R. M. Gray and J. T. Gill, "A Lagrangian formulation of fixed rate and entropy/memory constrained quantization," *DCC* 2005, March 2005

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

[4] A. K. Jain et al.,"Statistical Pattern Recognition: A Review," *IEEE Trans. PAMI.* 22(1), pp.4-37, 2000.

[5] A. Gersho and R. M. Gray, "Vector Quantization and Signal Compression," *Kluwer Academic Press*, 1992.

[6] J. H. Friedman, "Regularized Discriminant Analysis," *J. Am. Statistical Assoc.,* vol. 84, pp. 165-175. March 1989.

[7] A. K. Aiyer et al. "Lloyd Clustering of Gauss Mixture Models for Image Compression and Classification", *Signal Processing: Image Communication*, to appear.

[8] Sangho Yoon, "Regularizing Covariance Estimation by Quantized Eigenvalues and its Application to Image Classification," *Asilomar conference on Sig., Sys. and Comp.,* Nov, 2004, CA.

[9] Lloyd,S. "Least square quantization in PCM," *IEEE Transactions on Information Theory*, IT-28(2):129–137, March 1982.

[10] Mark J. F. Gales, "Maximum Likelihood multiple subspace projections for hidden Markov models Gales," Sppeech and Audio Processing, IEEE Trans. on Vol. 10, Issue 2, Feb. 2002.

[11] J. B. MacQueen,"Some methods for classification and analysis of multivariate observations," *5th Berkeley Symp. Math. Stat. Probabil*. Berkeley, CA: University of California Press, 1967

[12] Sugar, C. and James, G. "Finding the Number of Clusters in a Data Set : An Information Theoretic Approach," *Journal of the American Statistical Association* 98, 750-763, 2003.

[13] Fraley, C. and Raftery, A.E."How many clusters? Which clustering methods? Answers via model-based cluster analysis," *Computer Journal,* 41, 578-588. 1998.

[14] K. Fukunaga, "Introduction to Statistical Pattern Recognition", second ed., New York: *Academic Press*, 1990.

[15] J. G. Dy et al., "Feature Selection for Unsupervised Learning," *Journ. of Mach. Learn. Res.*, Vol. 5, August, 2004.

[16] Trevor Hastie et al.," The Elements of statistical learning," *Spring-Verlag*, 2001

[17] Dempster et al., "Maximum likelihood from incomplete data via the EM algorithm (with discussion)," *Journ. of the Royal Statis. Soc.*, Series B, Vol. 39, 1977, p. 1-38.

[18] Hardy, A. "On the number of clusters", *Computational Statistics and Data Analysis* 23, 1996.

[19] Mario A. T. et al., "Unsupervised Learning of Finite Mixture Models," *IEEE Trans. of PAMI*, Vol. 24, No.3, 2002.