ROBUST VECTOR QUANTIZATION BY COMPETITIVE LEARNING

Joachim M. Buhmann & Thomas Hofmann

Rheinische Friedrich–Wilhelms–Universität
Institut für Informatik III, Römerstraße 164, D-53117 Bonn, Germany
email:{jb,th}@cs.uni-bonn.de, http://www-dbv.cs.uni-bonn.de

ABSTRACT

Competitive neural networks can be used to efficiently quantize image and video data. We discuss a novel class of vector quantizers which perform *noise robust* data compression. The vector quantizers are trained to simultaneously compensate channel noise and code vector elimination noise. The training algorithm to estimate code vectors is derived by the maximum entropy principle in the spirit of *deterministic annealing*. We demonstrate the performance of noise robust codebooks with compression results for a teleconferencing system on the basis of a wavelet image representation.

1. INTRODUCTION

Vector quantization [6] deals with the problem of encoding an information source by means of a finite size codebook. If the code optimization is data driven, this is more specifically called *adaptive vector quantization*. Adaptive vector quantization possesses important applications, for example in speech and image compression, where the code has to be adapted to the statistics of the data source.

We present a novel approach to the design of optimal noise robust vector quantizers, which not only compactly encode the data, but also compensate channel noise and random code vector eliminations. *Robust vector quantization* is formulated as an optimization problem, extending the source-channel coding approach presented in [9, 5]. An efficient optimization heuristic is derived within the *deterministic annealing* framework [13, 3, 2]. The on-line version of the proposed algorithm is a neural network technique, which belongs to the class of *competitive learning* methods. The rigorous derivation from an optimization principle guarantees an information–theoretic interpretation of the presented algorithm. This also covers the self–organizing map [8] and the neural gas [12] as special cases.

2. ROBUST VECTOR QUANTIZATION

Assume a sample set of data vectors $\mathcal{X} = \{\mathbf{x}_i \in \mathbb{R}^d : 1 \leq i \leq N\}$ is given. *Adaptive* vector quantization requires to find an optimal codebook $\mathcal{Y} = \{\mathbf{y}_\alpha \in \mathbb{R}^d : 1 \leq \alpha \leq K\}$, with codebook vectors or prototypes $\mathbf{y}_\alpha$ such that the distortion induced by the data encoding is minimized. It is important to notice the distinction from computational learning theory, whether the distortion should be minimized only for the given data, or whether $\mathcal{X}$ is considered as training data to design a codebook for an on-line data generating source. Once a codebook $\mathcal{Y}$ and a corresponding encoding $e : \{1, \ldots, N\} \rightarrow \{1, \ldots, K\}$ has been specified, each data vector $\mathbf{x}_i$ is represented by its code vector $\mathbf{y}_{e(i)}$. The index $\alpha = e(i)$ is transmitted and $\mathbf{y}_\alpha$ is retrieved on the receiver side by codebook lookup. The information loss is measured by a distortion $\mathcal{D}(\mathbf{x}_i, \mathbf{y}_\alpha)$. The total distortion of encoding $\mathcal{X}$ with $\mathcal{Y}$ is given by,

$$\mathcal{H}^{\mathrm{vq}}(e, \mathcal{Y}) \quad = \quad \sum_{i=1}^{N} \mathcal{D}\left(\mathbf{x}_i, \mathbf{y}_{e(i)}\right). \tag{1}$$

For differentiable distortion measures, this results in the following set of stationary equations,

$$e(i) = \min\{\arg\min_{\nu} \mathcal{D}(\mathbf{x}_i, \mathbf{y}_\nu)\}, \tag{2}$$

$$\sum_{i=1}^{N} \delta_{e(i),\alpha} \frac{\partial}{\partial \mathbf{y}_\alpha} \mathcal{D}\left(\mathbf{x}_i, \mathbf{y}_\alpha\right) = 0. \tag{3}$$

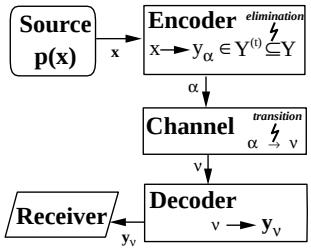
Eq. (2) is known as the *nearest neighbor rule* and Eq. (3) as the *centroid condition* from rate distortion theory. Squared Euclidean distortions imply the optimal choice of the codebook vectors as the center of mass of the associated data. Starting from Eqs. (2,3) different update schemes for reaching a local minimum are possible [10], moreover there exists a large number of heuristics to incrementally split clusters and to deal with unused codewords, c.f. [6].

An important extension of the vector quantization problem is to consider a noisy transmission channel in the codebook design phase. This problem is also known as *source-channel coding*. We assume the noise characteristics of the channel to be known and denote by $S_{\alpha\nu}$ the probability of receiving index ν after sending α through the channel. It has been noticed [9, 5, 11, 3] that source-channel coding may result in a topological ordering of prototypes, since the channel noise breaks

This work was supported by the Federal Ministry of Education and Science BMBF under grant # 01 M 3021 A/4.

the permutation symmetry of the prototype indices. A similar mechanism has also been applied in the self-organizing map [8].

A second fundamental extension of the basic vector quantization model deals with random eliminations of codebook vectors and is called *robust vector quantization* [7]. In this communication model the codebook design has to deal with the problem, that certain prototypes may not be available at encoding time t due to a temporary codebook reduction $\mathcal{Y}^{(t)} \subseteq \mathcal{Y}$. This might be necessitated by rapidly varying bandwidth limits, a problem also known as variable–rate vector quantization. In the biological context the $\mathbf{y}_\alpha$ may parameterize synaptic weights of neurons and eliminations of prototypes correspond to single neuron defects.



The possibility that only a part of the codebook is available to the encoder requires to specify a complete preference list or ranking of codebook vectors. For every data vector the ranking r_i is a bijective mapping on $\{1, \ldots, K\}$, representing a total ordering of codebook vectors. $r_i(\alpha) = 1, 2, \ldots, K$ denotes, that $\mathbf{y}_\alpha$ is the first, second, ..., K-th choice to encode $\mathbf{x}_i$. For the data encoding at time t the $\mathbf{y}_\alpha$ with the lowest rank $r_i(\alpha)$ among the available part $\mathcal{Y}^{(t)} \subseteq \mathcal{Y}$ of the codebook is used to encode $\mathbf{x}_i$. In the case of an independent elimination noise ϵ_α for $\mathbf{y}_\alpha$, the objective function for robust vector quantization is given by

$$\mathcal{H}^{\mathrm{rvq}}(r, \mathcal{Y}) = \sum_{i=1}^{N} \sum_{\alpha=1}^{K} (1 - \epsilon_\alpha) \left[\prod_{\nu, r_i(\nu) < r_i(\alpha)} \epsilon_\nu \right] D_{i\alpha}^S, \quad (4)$$

where $D_{i\alpha}^S = \sum_{\mu=1}^{K} S_{\alpha\mu} \mathcal{D}(\mathbf{x}_i, \mathbf{y}_\mu)$ is the expected distortion, for encoding $\mathbf{x}_i$ by index α. Notice, that for a given codebook the optimal choice for the rank variables is obtained by sorting $\mathcal{D}_{i\alpha}^S$ in ascending order.

In the simpler case without channel noise and with uniform elimination probabilities $\epsilon_\alpha = \epsilon$, $\forall \alpha$, the objective function is essentially equivalent to the *neural gas* model, introduced by Martinetz et al. [12].

3. CODEBOOK DESIGN BY DETERMINISTIC ANNEALING

In this section we apply the optimization framework called *deterministic annealing* (DA) for optimal codebook design. The core of all annealing methods is a computational temperature T, which controls the amplitude of noise, artificially introduced in the optimization process. At $T > 0$ the original cost function is replaced by a new effective cost function $\mathcal{F}_T$ called the *free energy*. In DA the free energy is minimized with respect to $\mathcal{Y}$ at every temperature level and the minimum is tracked while T is gradually lowered. In the high temperature regime $\mathcal{F}_T$ is convex, while we recover the original cost function in the limit of $T \to 0$. The free energy for vector quantization is given by [13],

$$\mathcal{F}_T^{\mathrm{vq}}(\mathcal{Y}) = -T \sum_{i=1}^{N} \log \sum_{\alpha=1}^{K} \exp\left[-\frac{1}{T} \mathcal{D}(\mathbf{x}_i, \mathbf{y}_\alpha) \right]. \quad (5)$$

If $\mathcal{D}$ is differentiable, the equations which result from minimizing the free energy are the generalized centroid conditions with Gibbs probabilities

$$\mathbf{P}\{e(i) = \alpha\} = \frac{\exp\left[-\frac{1}{T} \mathcal{D}(\mathbf{x}_i, \mathbf{y}_\alpha) \right]}{\sum_{\nu=1}^{K} \exp\left[-\frac{1}{T} \mathcal{D}(\mathbf{x}_i, \mathbf{y}_\nu) \right]}. \quad (6)$$

which replace the Kronecker delta functions $\delta_{e(i),\alpha}$ in Eq. (3). As can be seen easily, the nearest neighbor rule is a limiting case of the soft–min function in Eq. (6) for $T \to 0$.

For the robust model $\mathcal{H}^{\mathrm{rvq}}$ the application of the maximum entropy framework leads to an additional problem, which is the calculation of Gibbs averages for the encoding ranks. To obtain an efficient algorithm, we perform a *truncated deterministic annealing*. The idea is to apply the maximum entropy principle only to the lowest $r_{\max}$ rank variables of $\mathcal{H}^{\mathrm{rvq}}$ and to calculate higher ranks at $T = 0$. We will present the results for the simplest non–trivial case, $r_{\max} = 2$.

To facilitate the calculation, we impute the optimal ranks given indices α and ν with $r_i(\alpha) = 1$, $r_i(\nu) = 2$, by ordering the remaining indices $\mu \neq \alpha, \nu$ with ascending $D_{i\mu}^S$. $h_{\alpha\nu}^i$ denotes the *residual costs* for encoding $\mathbf{x}_i$, after $\mathbf{y}_\alpha$ and $\mathbf{y}_\nu$ have been eliminated. We obtain expected distortions

$$\phi_{\alpha\nu}^i = (1 - \epsilon_\alpha) D_{i\alpha} + (1 - \epsilon_\nu) \epsilon_\alpha D_{i\nu} + \epsilon_\alpha \epsilon_\nu h_{\alpha\nu}^i. \quad (7)$$

From these we calculate the Gibbs probabilities

$$\mathbf{P}\{r_i(\alpha) = 1 \wedge r_i(\nu) = 2\} = \frac{\exp\left[-\frac{1}{T} \phi_{\alpha\nu}^i \right]}{\sum_{\gamma=1}^{K} \sum_{\mu \neq \gamma} \exp\left[-\frac{1}{T} \phi_{\gamma\mu}^i \right]}, \quad (8)$$

which we abbreviate by $p_{\alpha\nu}^i$ in the sequel. Further denote by r_i^* the optimal ranking. Marginalization allows us to calculate the encoding probabilities,

$$\begin{aligned} \mathbf{P}\{e(i) = \alpha\} &= (1 - \epsilon_\alpha) \sum_{\nu=1}^{K} \left(p_{\alpha\nu}^i + \epsilon_\nu p_{\nu\alpha}^i \right) \quad (9) \\ &+ (1 - \epsilon_\alpha) \sum_{\nu \neq \alpha} \sum_{\mu \neq \nu, \alpha} p_{\nu\mu}^i \epsilon_\nu \epsilon_\mu \prod_{\substack{\gamma, r_i^*(\gamma) < r_i^*(\alpha) \\ \gamma \neq \nu, \mu}} \epsilon_\gamma \end{aligned}$$

The received index $d(i)$ to decode $\mathbf{x}_i$ is a random variable and the decoding probabilities $\mathbf{P}\{d(i) = \alpha\}$ are the weights for the centroid conditions in Eq. (3),

$$\mathbf{P}\{d(i) = \alpha\} = \sum_{\nu=1}^{K} S_{\nu\alpha} \mathbf{P}\{e(i) = \nu\}. \quad (10)$$

	Level 1			Level 2			Level 3		
	dtl x	dtl y	dtl xy	dtl x	dtl y	dtl xy	dtl x	dtl y	dtl xy
[Blocks]	16	16	-	4	4	16	4	4	4
[Codebook]	64	64	0	128	128	256	256	256	128
mbpp	6/16	6/16	0	7/4	7/4	4/16	8/4	8/4	7/4

Table 1: Block, codebook sizes and maximum bits per pixel (mbpp) for all wavelet detail images (dtl x,y,xy).

4. COMPETITIVE LEARNING

For many interesting applications, it is more adequate to consider an on-line setting, where solutions are adapted sequentially with the presentation of new data. A systematic way to obtain on-line equations for squared Euclidean distortions is to approximate the difference $\mathbf{y}_\alpha^{N+1} - \mathbf{y}_\alpha^N$ between the centroids for N and $N+1$ data vectors [3]. The resulting equations are given by

$$\mathbf{y}_\alpha^{N+1} = \mathbf{y}_\alpha^N + \frac{\mathbf{P}\{d(N+1)=\alpha\}}{p_\alpha^{N+1}} \left(\mathbf{x}_{N+1} - \mathbf{y}_\alpha^N\right), \quad (11)$$

where $p_\alpha^{N+1} = p_\alpha^N + \mathbf{P}\{d(N+1)=\alpha\}$ is a running average for prototype $\mathbf{y}_\alpha$, $p_\alpha^0 = 1$.

On–line update schemes for vector quantization are closely related to algorithms known as *competitive learning* in the neural networks community. Prototypes correspond to neuron weights, which determine the center of the receptive field in the data or stimuli space. If neurons are activated, they get tuned to that specific stimuli, a relation which is directly represented in Eq. (11), since the direction of weight changes is always towards the new stimulus $\mathbf{x}_{N+1}$. Compared to the on–line learning rules proposed by Kohonen et al. [8] and Martinetz et al. [12], the *learning rate* in Eq. (11) is different for every 'neuron', since it depends inversely on the number of data points assigned so far. This dependency on the history of a neuron avoids the problem of defining an appropriate global learning rate. To accelerate the convergence rate it might be advantageous to include an additional learning gain, based on the 'Search–Then–Converge' heuristic [4]. The modified update rules are obtained by replacing p_α^{N+1} by $1 + p_\alpha^{N+1}/\eta$. For $\eta > 1$, the mobility of the neuron weights $\mathbf{y}_\alpha$ is increased.

5. RESULTS

We have tested the presented vector quantization algorithm on wavelet-transformed video sequences from a teleconferencing application. Due to the fact, that severe bandwidth limitations as well as noisy transmission channels are a typical problem especially for wireless teleconferencing, this is a realistic scenario for robust vector quantization. The wavelet transformation and the grouping scheme of wavelet coefficients into blocks is due to [1], with a three-level biorthogonal wavelet transformation. Data vectors were generated by grouping neighboring wavelet coefficients in blocks of size 4×4 and 2×2 for each sub-band separately, c.f. Table 1.

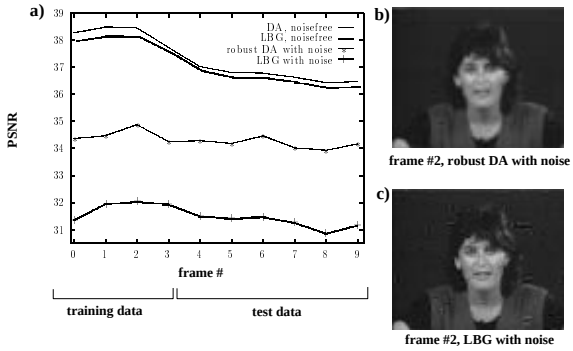


Figure 1: Batch optimization on the 'Miss America' sequence (354×288 pixels). (a) PSNR curve for the noise–free and noisy case ($\epsilon_\alpha = \epsilon = 0.5$, bit-noise 1%) with codebooks trained by LBG and DA. (b),(c) example frames for the noisy case.

This results in a multi-resolution codebook with independent codebooks for all wavelet detail signals. The residuum on the third level was not further vector quantized, in practice a predictive coding combined with a scalar quantization showed the best compression performance. Results for batch optimization are depicted in Fig. 1. Codebooks designed by DA not only yield superior results on the training data, but also on new test data, as compared to the LBG algorithm. The robust codebook design shows an improvement of approximately 3 dB on both, training and test data. This means 50% of the PSNR loss due to noise are compensated by the robust design procedure. In the noisefree case all codebooks obey a significant difference of about 2 dB in PSNR between the training and the test error due to data overfitting, while no overfitting phenomenon occurs in the presence of noise. Without noise the improvement by DA is approximately 0.2 dB.

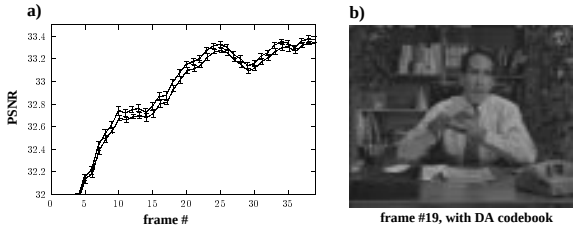


Figure 2: On-line learning with and without temperature. (a) PSNR at $T > 0$ (upper curve) and $T = 0$ (lower curve), averaged over 20 runs with $\eta = 30$ and an average entropy of 0.5 for the $T > 0$ experiments. (b) example frame of 'Salesman' sequence.

Results for noisefree on-line learning are depicted in Fig. 2. The rapidly increasing PSNR curves in (a) demonstrates a fast adaptation to the statistics of the source with a final quality of about 33.5 dB PSNR. The experiments with a non–zero temperature $T > 0$ demonstrate the non–trivial fact, that the introduction

of temperature improves the on–line learning performance. We have implemented an on–line temperature control, which increases or decreases the temperature, such that a prespecified average entropy is obtained. Since in on–line learning the distortion is always determined on new data, this means, that a non–zero temperature yields a better generalization as compared to the WTA.

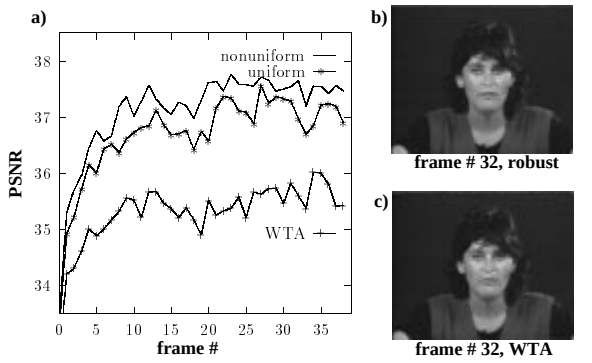
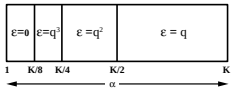


Figure 3: On-line learning with codebook vector eliminations on the 'Miss America' sequence. (a) PSNR curves for robust vector quantization with $q = 0.685$ (upper curve) and $\epsilon = 0.5$ (middle), compared to WTA (lower curve). (b),(c) example frames for robust encoding with $q = 0.685$ and WTA, respectively.



A series of experiments with elimination noise are summarized in Fig. 3. The gain achieved by robust codebook design is more than 2 dB. We have investigated the full robust model with elimination probabilities which depend on the prototype index α, according to the depicted scheme. The elimination probabilities for half of the codebook vectors are high, $\epsilon_\alpha = q$, and are recursively reduced, until a 'core codebook' is obtained, which has zero elimination probability. As opposed to a uniform elimination probability, this will partially break the permutation symmetry of codewords. Our experiments clearly indicate that the codebook design can take advantage of this knowledge if compared to a uniform elimination model with the same average elimination probability. This is also a more realistic assumption for fast on-line rate control, since it is not recommendable to eliminate codebook vectors completely at random. This demonstrates the advantage of the flexibility of the robust vector quantization framework in realistic encoding and transmission situations, taking into account possible sources of additional uncertainty.

6. CONCLUSION

Robust vector quantization is a lossy data compression technique which compensates for channel and code vector elimination noise. We have derived neural network algorithms with underlying competitive learning schemes both for batch and on-line learning. The resulting vector quantizers have proven to yield superior quantization results compared to standard design techniques.

Acknowledgment: It is a pleasure to thank H. Klock and A. Polzer for significant support and help with the teleconferencing experiments.

7. REFERENCES

[1] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies. Image coding using wavelet transform. *IEEE Transactions on Image Processing*, 1(2):205–220, 1992.

[2] J. M. Buhmann and H. Kühnel. Complexity optimized data clustering by competitive neural networks. *Neural Computation*, 5:75–88, 1993.

[3] J. M. Buhmann and H. Kühnel. Vector quantisation with complexity costs. *IEEE Transactions on Information Theory*, 39:1133–1145, 1993.

[4] Ch. Darken and J. Moody. Note on learning rate schedules for stochastic optimization. In *Advances in Neural Information Processing Systems*, volume 4, 1991.

[5] M. Farvardin. A study of vector quantization for noisy channels. *IEEE Transactions on Information Theory*, 36:799–809, 1990.

[6] A. Gersho and R. M. Gray. *Vector Quantization and Signal Processing*. Kluwer Academic Publisher, Boston, 1992.

[7] Th. Hofmann and J.M. Buhmann. An annealed neural gas network for robust vector quantization. In *ICANN'96*, Lecture Notes in Computer Science 1112. Springer Verlag, 1996.

[8] T. Kohonen. Self-organizing formation of topologically correct feature maps. *Biological Cybernetics*, 43:59–69, 1982.

[9] H. Kumazawa, M. Kasahara, and T. Namekawa. A construction of vector quantizers for noisy channels. *Electronics and Engeneering in Japan*, 67B(4):39–47, 1984.

[10] Y. Linde, A. Buzo, and R. M. Gray. An algorithm for vector quantizer design. *IEEE Transactions on Communications*, 28:84–95, 1980.

[11] S.P. Luttrell. Hierarchical vector quantization. *IEE Proceedings*, 136:405–413, 1989.

[12] T. Martinetz, S.G. Berkovich, and K.J. Schulten. Neural-gas network for vector quantization and its application to time-series prediction. *IEEE Transactions on Neural Networks*, 4(4):558–569, 1993.

[13] K. Rose, E. Gurewitz, and G. Fox. Vector quantization by deterministic annealing. *IEEE Transactions on Information Theory*, 38(4):1249–1257, 1992.