

Joint Optimization of Distributed Broadcast Quantization Systems for Classification

Michael A. Lexa and Don H. Johnson
Department of Electrical and Computer Engineering
Rice University, Houston, TX 77251-1892
amlexa@rice.edu, dhj@rice.edu

Abstract

We develop a simulated annealing technique to jointly optimize a distributed quantization structure meant to maximize the asymptotic error exponent of a downstream classifier or detector. This distributed structure sequentially processes an input vector and exploits broadcasts to improve the best possible error exponents. The annealing approach is a robust technique that avoids local maxima and is easily tailored to a broadcast quantizer's structural constraints.

1. Introduction

Quantization for classification concerns a class of problems where a quantizer serves as a preprocessor to a downstream classifier. *Distributed* quantization for classification is an extension of this idea where, instead of a single quantizer, a collection of quantizers acts as a preprocessor. The single quantizer version of this problem was introduced in the late 1970's [5, 11], and in the intervening years, various authors have touched on various aspects of the distributed version [1, 9], yet quantization for classification, along with the more general notion of distributed task-driven quantization, still remains rather underdeveloped compared to quantization theory as a whole [3]. Here we discuss a distributed quantization system that exploits information shared among the constituent quantizers [8] and an optimization technique that optimizes the performance of the follow-on classifier. In particular, we describe a distributed structure that broadcasts information as the quantizers sequentially process an input vector and propose an efficient and robust algorithm that jointly optimizes the quantizers so as to maximize the best possible asymptotic error exponent of the classifier.

Despite the fact that our problem resembles typical distributed source coding (quantization) formulations, results and techniques from those problems surprisingly have little bearing on our problem. Because we optimize asymptotic error exponents, our optimizations do not involve reconstruction points or codewords, nor do the partitions induced by our quantizers obey the nearest neighbor rule. Thus the Lloyd optimality conditions [2] are not applicable and consequently, neither are algorithms designed to satisfy them, even those recently proposed for distributed settings (see e.g. [12, 14]). Here we instead develop a simulated annealing algorithm [6] to solve the optimization. This algorithm is not only robust to initializations and not easily trapped in local minima or maxima, but it is very amenable to the *structural and functional constraints* imposed by a broadcast system, i.e. the constraints are easily incorporated and enforced in the optimization.

2. System structure and asymptotic error exponents

2.1. Broadcast quantizers

A vector quantizer on \mathbb{R}^M is a function α that maps a real number into an index set: $\alpha : \mathbb{R}^M \mapsto \{0, \dots, L^M - 1\}$. If the real-valued random vector \mathbf{X} serves as an input to α and $\mathbf{Y} \in \{0, \dots, L^M - 1\}$ represents its quantized output, then we have $\mathbf{Y} = \alpha(\mathbf{X})$. A *distributed broadcast quantizer* ϕ is a vector quantizer composed of a collection of M spatially separated component quantizers $\{\phi_1, \dots, \phi_M\}$ that obey,

$$Y_m = \begin{cases} \phi_m(X_m, \mathbf{Y}_1^{m-1}), & 1 \leq m \leq K \\ \phi_m(X_m, \mathbf{Y}_{m-K}^{m-1}), & K < m \leq M, \end{cases} \quad (1)$$

where $Y_m \in \{0, \dots, L - 1\}$, $K \in \{0, \dots, M - 1\}$, $\mathbf{Y}_{m-K}^{m-1} = (Y_{m-K}, \dots, Y_{m-1})$, and $X_m \in \mathbb{R}$. Each component quantizer ϕ_m thus maps a real-valued random variable plus at most K values drawn from the set $\{0, \dots, L - 1\}$ into $\{0, \dots, L - 1\}$.

The functional relationships in (1) imply a sequential ordering in how a broadcast structure quantizes its input $\mathbf{X} = (X_1, \dots, X_M)$. First, ϕ_1 quantizes X_1 and then broadcasts its output y_1 . Next, ϕ_2 quantizes X_2 but takes into account y_1 , and then broadcasts its output. The process continues until all M component quantizers produce an output with each succeeding quantizer basing its output on its input and the K preceding outputs. Because the component quantizers for $K > 0$ react to past outputs, the system is *dynamic*. That is, partial information is shared among the quantizers as they sequentially process the components of the input vector \mathbf{X} . Broadcast quantizers belong to the general class of sequential quantizers [2], but here we use the term “broadcast” to emphasis how each component quantizer communicates.

The parameter K determines the “depth” to which outputs are broadcast. If $K = M - 1$, then for every m , each quantizer broadcasts its output to all remaining quantizers ϕ_k , $k > m$. If $K = 0$, there are no inter-quantizer communications and the resulting structure is static or *noncommunicative*.

Note that with the functional dependency on \mathbf{Y}_{m-K}^{m-1} , the component quantizers ϕ_m , $m = 1, \dots, M$ are not scalar quantizers in the traditional sense because they are not characterized by a single mapping from the reals to an index set. Rather, they are collections of scalar quantizers since for every realization \mathbf{y}_{m-K}^{m-1} of the random vector \mathbf{Y}_{m-K}^{m-1} , we can associate a scalar quantization rule $\phi_{m,j}$ such that $Y_m = \phi_{m,j}(X_m)$, where $\phi_{m,j}(X_m)$ is shorthand notation for $\phi_m(X_m, \mathbf{y}_{m-K}^{m-1})$ and j is an index for the L^K different outcomes of \mathbf{Y}_{m-K}^{m-1} . Thus a full description of ϕ_m , for any given $m > K$, requires the specification of L^K scalar quantizers $\{\phi_{m,0}, \dots, \phi_{m,L^K-1}\}$. Overall, therefore, a broadcast quantizer ϕ can be viewed either as a collection of M component quantizers each satisfying (1) or as a large set of scalar quantizers $\{\phi_1, \phi_{m,j}, m = 2, \dots, M, j = 0, \dots, L^{m-1}, m < K; j = 0, \dots, L^K - 1, m \geq K\}$.

Because the quantizers $\phi_{m,j}$ are all scalar quantizers, each can also be described by a labeled partition of the real line. Thus for fixed m and j , a partition associated with $\phi_{m,j}$ is characterized by a threshold vector $\tau_{m,j} = (\tau_{m,j}^{(0)}, \dots, \tau_{m,j}^{(N_{m,j})})$ whose elements define $N_{m,j}$ intervals,

$$I_{m,j}^{(0)} = (-\infty, \tau_{m,j}^{(1)}], \quad I_{m,j}^{(1)} = [\tau_{m,j}^{(1)}, \tau_{m,j}^{(2)}], \quad \dots \quad I_{m,j}^{(N_{m,j}-1)} = [\tau_{m,j}^{(N_{m,j}-1)}, \infty). \quad (2)$$

To each interval one index or label is assigned from the set $\{0, \dots, L - 1\}$. Taken together, this partition and its labeling is a complete characterization of $\phi_{m,j}$. For a fixed $m > K$, the

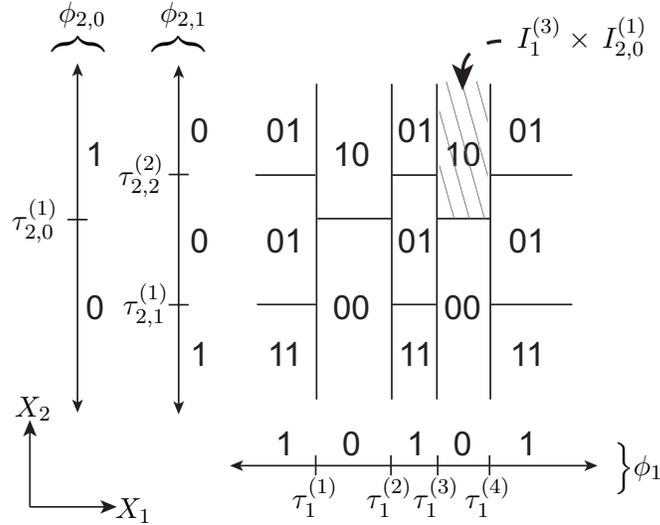


Figure 1. A hypothetical broadcast quantizer ϕ depicted as a labeled partition for $M = 2, L = 2, K = 1$.

L^K partitions of \mathbb{R} defined by the set of threshold vectors $\{\tau_{m,0}, \dots, \tau_{m,L^K-1}\}$, along with their associated labelings, explicitly characterize the component broadcast quantization rule ϕ_m .

A broadcast quantizer ϕ can also be described by a labeled partition that is formed by taking Cartesian products of the intervals in (2) and by concatenating their corresponding labels. For example, Figure 1 shows a hypothetical partition and labeling of \mathbb{R}^2 ($M = 2, L = 2, K = 1$). The broadcast quantization rule ϕ_1 is displayed horizontally along the X_1 axis, and $\phi_{2,0}$ and $\phi_{2,1}$ are displayed vertically along the X_2 axis. The hatched cell is the product $I_1^{(3)} \times I_{2,0}^{(1)}$ and its label 10 is the concatenation of their labels. Note that all the cells are rectangular because broadcast quantizers process their inputs in a distributed manner according to (1). In fact, regardless of the parameter values M, L, K , every quantizer $\phi_{m,j}$ can only operate on a single dimension of \mathbf{X} . Thus ϕ 's partitioning cells always have hyper-rectangular geometry. We refer to this induced structure as the structural constraints of ϕ .

From a different perspective, Figure 1 shows the additional degrees of freedom in ϕ brought by broadcasting. Unlike a comparable noncommunicative structure, ϕ_2 consists of two constituent scalar quantizers $\phi_{2,0}, \phi_{2,1}$. Thus one can either take the Cartesian product of ϕ_1 with $\phi_{2,0}$, or with $\phi_{2,1}$ to form the columns of ϕ . In Figure 1, this degree of freedom manifests itself as “misaligned” rows with labels that change across the columns. Such behavior is not exhibited in noncommunicative structures whose overall partition of \mathbb{R}^M resembles that of standard vector product quantizers.

2.2. Kullback-Leibler divergence

The Kullback and Leibler divergence is an information theoretic quantity that quantifies the “dissimilarity” between two probability distributions and is defined as follows. Suppose P and Q are probability measures on a measurable space $(\mathcal{X}, \mathcal{L})$ and suppose that they are absolutely continuous with respect to one another. Then, letting $p(x)$ and $q(x)$ denote the

density functions of P and Q , the Kullback-Leibler (KL) divergence between $q(x)$ relative to $p(x)$ is defined as [7],

$$\mathcal{D}(p\|q) \triangleq - \int_{\mathcal{X}} p(x) \log \frac{q(x)}{p(x)} dx. \quad (3)$$

The choice of the logarithm's base is arbitrary, but here we use base two. When the outcome space \mathcal{X} is discrete, the integral in (3) becomes a summation.

Our use of the divergence as a quantity to be optimized is motivated by Stein's Lemma [7], a well-known result that relates the divergence to the asymptotic performance of an optimal detector. In words, it says that the error probabilities of an optimal Neyman-Pearson detector decay exponentially as the number of observations increase and that the limiting exponential rate is equal to the divergence between the distributions characterizing the detector's inputs. Therefore, by optimizing the divergence at the output of the broadcast system, we maximize the best possible asymptotic error decay rate of the follow-on classifier.

Though motivated in part by Stein's Lemma, more important for the current discussion is the fact that the divergence obeys the Data Processing Theorem.¹ This result effectively states that no processing can increase the discrimination information (i.e., the divergence) originally present in a system's inputs [7]. Thus, no follow-on detector can recover from a divergence loss resulting from a distributed quantization process. It is therefore critical to design the distributed system well and not rely on sophisticated detection algorithms. For this reason, we investigate here broadcast quantization structures in contrast to noncommunicative ones.

Now, suppose the input vector \mathbf{X} is distributed in one of two ways: $H_0 \sim p_{\mathbf{X}}$, $H_1 \sim q_{\mathbf{X}}$, where here $p_{\mathbf{X}}$ and $q_{\mathbf{X}}$ denote the joint input probability densities under the two hypotheses $H_j, j = 0, 1$. Then, denoting the joint output probability mass functions induced by ϕ by the vectors

$$\begin{aligned} p(\phi) &= (p_0(\phi), \dots, p_{L^M-1}(\phi)) && \text{under } H_0 \\ q(\phi) &= (q_0(\phi), \dots, q_{L^M-1}(\phi)) && \text{under } H_1, \end{aligned} \quad (4)$$

where the probabilities $p_i(\phi)$ under H_0 equal $\Pr(\phi(\mathbf{X}) = i; H_0)$, $i = 0, \dots, L^M - 1$, and similarly for $q_i(\phi)$ under H_1 , the problem of interest may be expressed as

$$\max_{\phi} \mathcal{D}(p(\phi)\|q(\phi)). \quad (5)$$

One can think of (5) as either a problem of determining one optimal vector quantizer ϕ , or as a problem of determining M one-dimensional component quantizers $\phi_{m,j}$. The next section presents a solution method based on simulated annealing.

3. Simulated annealing with fixed partitions

Simulated annealing is a well-known stochastic relaxation optimization technique that mimics the metallurgical process of annealing, where a metal (steel, for instance) is heated and then slowly cooled [6, 15]. The slow rate of cooling allows the molecules within the metal to crystallize in a low-energy equilibrium state, rendering the metal less brittle and more workable. When this idea is applied in optimization problems, the objective function is "slowly cooled" so as to prevent solutions from being trapped in local minima or maxima.

¹Kullback refers to the Data Processing Theorem as the divergence's *invariance property* [7, pp. 18-22].

In practice one simulates this behavior by randomly perturbing the state of a finite discrete set over which the objective function is defined, and say for a maximization, accepting perturbations that *decrease* the objective function with some *probability* instead of simply rejecting them. The probabilities in a simulated annealing algorithm primarily depend on what is known as a *temperature schedule* which is simply a sequence of decreasing numbers $\{T_n\}$ that controls the rate at which these probabilities decrease. Typically the probabilities are given by the formula $\exp\{-|\Delta|/T_n\}$ where Δ is the difference in the objective function's value in a current versus a perturbed state and $n = 1, 2, \dots$ [15]. For any given temperature, the objective function is allowed to reach what is qualitatively equivalent to an equilibrium state for that temperature. As the temperature decreases, the algorithm converges in probability under suitable conditions to a globally optimal solution [15]. This convergence result holds broadly; however, it is often difficult in practice to determine whether the algorithm has indeed converged and in some cases, the time to convergence can be exponentially slow [10].

In (5), we posed the problem of maximizing the output divergence as a functional optimization problem—find the quantization rule ϕ on \mathbb{R}^M that maximizes $\mathcal{D}(p(\phi)||q(\phi))$ and obeys a broadcast system's structural constraints. However to employ simulated annealing, the problem must be re-formulated since this technique is geared towards combinatoric problems, not functional optimizations. In a nutshell, the re-formulation specifies a partition of \mathbb{R}^M a priori and then use the annealing algorithm to search for an optimal labeling of that partition. Thus instead of attempting to determine an optimal ϕ outright, we impose a partition that conforms to a broadcast system's inherent structural constraints and then solve the remaining combinatoric labeling problem.

The form of this partition clearly plays an important role in the efficacy of the algorithm, for a naive choice could severely degrade the quality of the resulting estimate $\hat{\phi}$, regardless of the cells' labelings. A convenient choice (because of its simplicity) is to uniformly tessellate a *bounded* volume of \mathbb{R}^M with hyper-cubes where the number of partitioning cells is greater than the size of ϕ 's output alphabet, i.e. larger than L^M . Observe that such a partition, which we shall denote by \mathcal{S} , is consistent with the inherent structural constraints of a distributed broadcast system. In fact, one can form a bounded uniform tessellation simply by setting the number of thresholds $N_{m,j}$ in all component rules $\phi_{m,j}$ equal and requiring all the thresholds within the bounded region to be equally spaced.

Each cell's label in \mathcal{S} represents membership in one of the L^M partitioning regions (cells) R_i associated with ϕ . Formally, we can define these regions as unions of like-labeled cells in \mathcal{S} ,

$$R_i = \bigcup_{\substack{k:l(S_k)=l(R_i) \\ k \in \mathbb{Z}^+}} S_k, \quad i \in \{0, \dots, L^M - 1\}, \quad (6)$$

where S_k denotes a cell in \mathcal{S} and $l(\cdot)$ refers to the label of its argument. The annealing process therefore attempts to find the best clustering of labels to produce the estimate $\hat{\phi}$. Besides a label, each cell in \mathcal{S} has two probabilities associated with it which are the probability measures of the cell under H_0 and H_1 . The probabilities $p_i(\phi)$ and $q_i(\phi)$ of the output distributions $p(\phi)$ and $q(\phi)$ are computed by summing the probabilities of all like-labeled cells in \mathcal{S} , i.e., $p_i(\phi) = \Pr(R_i; H_0)$, $q_i(\phi) = \Pr(R_i; H_1)$.

At each iteration the annealing algorithm perturbs the labeling of \mathcal{S} , computes the divergence between the output distributions induced by the perturbed labeled partition, and probabilistically updates the labels according to the annealing process. The algorithm ceases when a stopping criterion is met and the final labeled partition \mathcal{S} yields the estimate

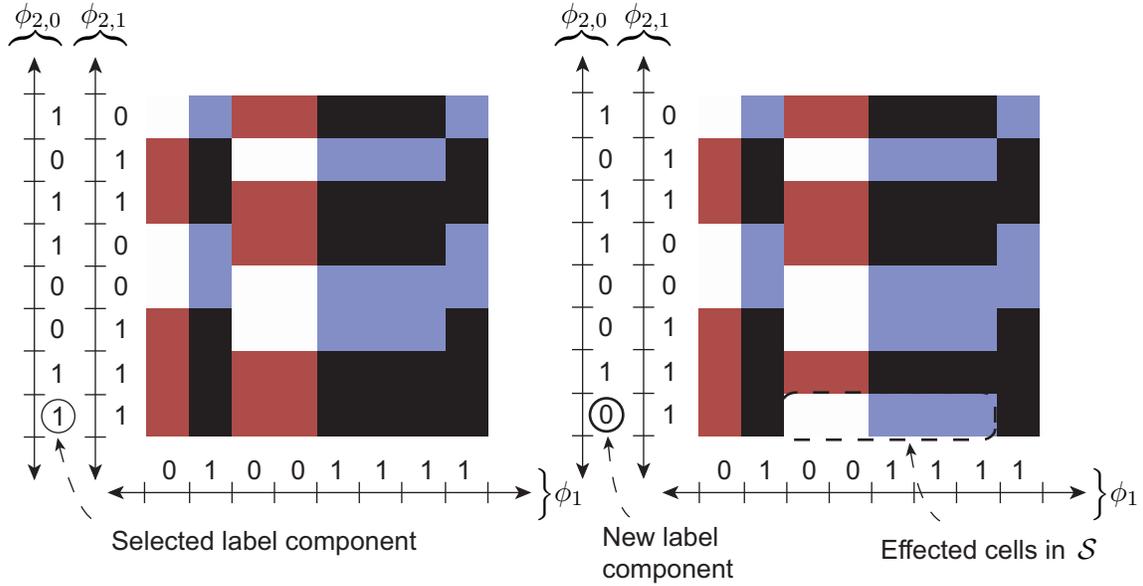


Figure 2. Example label component perturbation when \mathcal{S} is an 8×8 grid. The left panel shows an intermediate labeling of \mathcal{S} in the annealing process. The right panel shows the resulting labeling after a label component perturbation. Note that when the component label in $\phi_{2,0}$ (circled 1) is changed to a zero, five cells in \mathcal{S} associated with $\phi_{2,0}$ are affected.

$\hat{\phi}$. Details about the perturbations, stopping criterion, and the temperature schedules are given in the following sections.

Note that even though simulated annealing algorithms theoretically converge to a globally optimal solution for combinatoric problems, our implementation yields only an estimate of the true globally optimal broadcast quantization rule. First of all, the imposition of \mathcal{S} discretizes the continuous problem of (5). Thus, even if the annealing algorithm converged to a globally optimal solution in the discretized (combinatoric) problem, this solution would generally only approximate the actual optimal broadcast quantizer. Second, as noted above, it is often difficult to determine if the algorithm has indeed converged. Because of this obstacle, our implementation uses a heuristic stopping criterion that does not formally verify convergence.

3.1. Perturbations

During the annealing process, all perturbations must conform to the structural constraints of the distributed problem, otherwise one runs the risk that the resulting quantizer is not realizable in a distributed setting. In a centralized scenario, where there are no structural constraints, a perturbation simply consists of randomly selecting a cell in \mathcal{S} and randomly changing its label. Annealing with such perturbations would ultimately yield a ϕ that would require a system to have complete access to all inputs X_1, \dots, X_M . Thus, to ensure feasible solutions, perturbations in our algorithm alternate between one of two types: a *label component* and a *membership* perturbation. A label component perturbation begins with a random cell selection in \mathcal{S} and is then followed by a random selection of one of the intervals defining the cell. For example, suppose a cell defined by $I_1^5 \times I_{2,0}^6$ was randomly

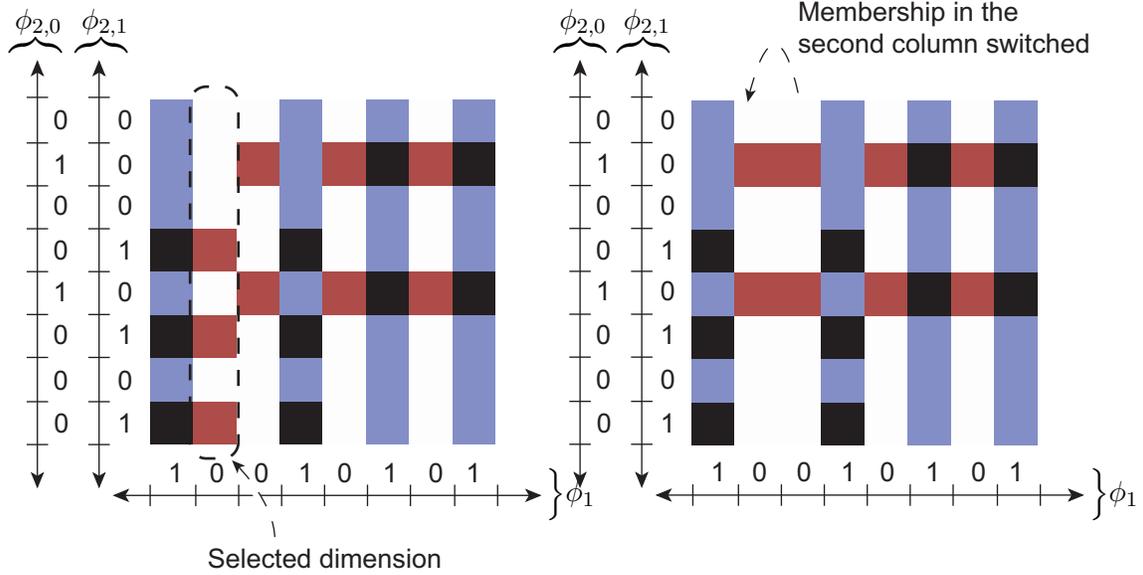


Figure 3. Example membership perturbation when \mathcal{S} is an 8×8 grid. The left panel shows an intermediate labeling of \mathcal{S} in the annealing process and the randomly selected dimension (here X_2 's dimension). The right panel shows the resulting labeling after a membership perturbation. Note that the second column of \mathcal{S} in the right panel is now associated with $\phi_{2,0}$ instead of $\phi_{2,1}$.

chosen and $I_{2,0}^6$ was selected among the pair. Next, a new label from the set $\{0, \dots, L-1\}$ is randomly generated and assigned to the selected interval (in this case $I_{2,0}^6$). Finally, in adherence to the structural constraints this label component change must also be made to all other cells in which the selected interval is a factor in its Cartesian product. Denoting a generic interval by $I_{m,j}^{(k)}$, the initial random cell selection fixes j and in doing so, effectively chooses one of the L^K component rules comprising ϕ_m . The random interval selection fixes m , the dimension of interest. Figure 2 shows one instance of a label perturbation when $M = 2, L = 2$, and $K = 1$.

Similar to a label perturbation, a membership perturbation involves a random cell selection in \mathcal{S} and a random selection of an interval (or equivalently a random choice of dimension). But a membership perturbation also includes a second random cell selection and the label assignments, $l(I_{m,j_1}^{(k)}) = l(I_{m,j_2}^{(k)})$, for $k = 0, \dots, N-1$, where m, j_1 , and j_2 are all determined by the two random cell selections. The overall effect therefore is to replace the quantization rule ϕ_{m,j_1} with ϕ_{m,j_2} in the randomly chosen dimension m . Figure 3 illustrates an example.

3.2. Temperature schedule and stopping criterion

At each temperature, the algorithm must satisfy a stopping criterion before moving onto the next temperature. In accord with the annealing analogy, the stopping criterion should mark the points when “equilibrium states” are reached. Here, like [4], we tie the notion of equilibrium to the rate of accepting perturbations (those accepted either because they produce an increase in divergence or those that decrease it but are probabilistically accepted). The *acceptance rate* is periodically calculated, and its empirical variance is then compared

to a threshold.

$$\text{acceptance rate} = \frac{\text{no. of times perturbations are accepted in a period at temp } T_n}{\text{total no. of iterations in period at temp } T_n}$$

When the threshold is exceeded, iterations at the current temperature stop and the algorithm either proceeds to anneal at the next temperature or stops altogether if the index n reaches its maximal value J . The number of temperature stages is determined beforehand and held fixed throughout the process.

As the algorithm progresses through its temperature schedule, the acceptance rate tends to decrease for two reasons. First, it becomes less likely that a perturbation will increase the divergence as the labeling improves and second, lower temperatures reduce the probability of accepting perturbations that decrease the divergence. Thus the set of threshold values used in the stopping mechanism is a non-increasing sequence whose length equals the number of temperature stages. For the example presented below, threshold values for the standard deviation of the acceptance rate ranged from 10^{-2} to 10^{-3} .

Unfortunately, as with any application of the simulated annealing algorithm, setting a temperature schedule is problem dependent. Through experimentation we found that the schedule, reported in the vector quantization literature [2, p. 369], was useful for the range of problems considered: $T_n = T_0(1 - n/J)^3$, $n = 1, \dots, J$, where T_0 represents an initial temperature.²

The difficulty in finding appropriate schedules is a commonly voiced drawback to simulated annealing. An approach such as deterministic annealing [13, 14] is appealing in this regard because it completely eliminates the finicky dependence on temperature schedules. However, it is not obvious (at least to the authors) that a broadcast system's structural constraints can be successfully enforced in the deterministic annealing setting.

4. Numerical example

Consider a simple broadcast structure composed two component quantizers with a single broadcast link from ϕ_1 to ϕ_2 ($M = 2, K = 1$). Let each quantizer have a two-element output index set ($L = 2$) and suppose the inputs X_1 and X_2 are jointly Gaussian $\mathcal{N}(\mathbf{m}_j, \Sigma_j)$, $j = 0, 1$, under both hypotheses, where

$$H_0 : \mathbf{m}_0 = (0 \ 0), \Sigma_0 = \begin{pmatrix} 1 & 0 \\ 0 & 1/3 \end{pmatrix}; \quad H_1 : \mathbf{m}_1 = \left(\frac{1}{4} \ \frac{1}{4}\right), \Sigma_1 = \begin{pmatrix} 1 & \rho\sqrt{1/3} \\ \rho\sqrt{1/3} & 1/3 \end{pmatrix}.$$

Letting \mathcal{S} be a 64×64 grid of cells and ρ (correlation coefficient) be 0.9, the algorithm stepped through seven temperatures and ceased after 17000 iterations when the variance of the acceptance rate dropped below 10^{-3} . The left panel in Figure 4 shows the initial random labeling of \mathcal{S} and the right panel shows the final labeling, or equivalently the final estimate $\hat{\phi}$. The final output divergence $\mathcal{D}(p(\hat{\phi})\|q(\hat{\phi}))$ equals 1.4897 bits. Compared to an optimized noncommunicative system (with a output divergence equal to 0.7809 bits), broadcasting improved the asymptotic error exponent of the follow-on detector by approximately 91%. By using our annealing algorithm we find that in this case a broadcast structure achieves significantly better performance than a noncommunicative one.

²In our implementation we actually use two temperature schedules—one associated with label component perturbations and the other with membership perturbations. Both have the same form but differ in their initial values.

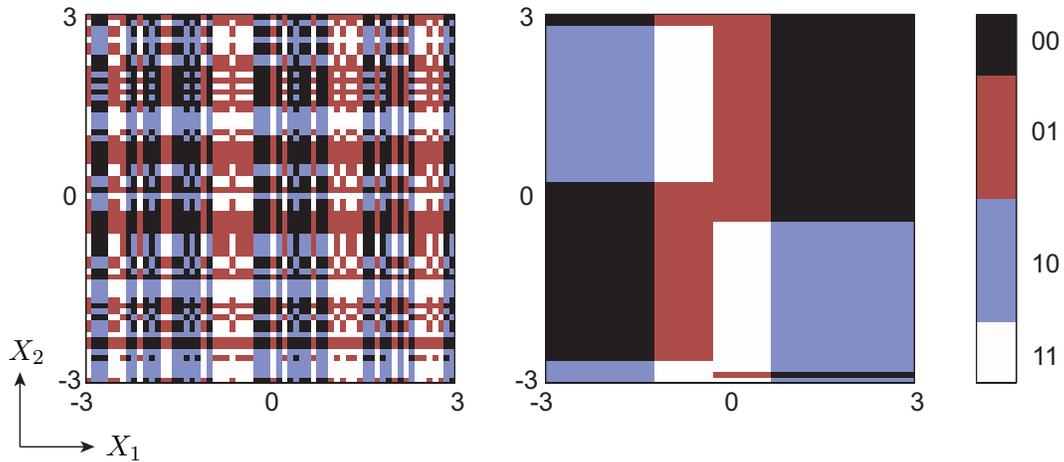


Figure 4. Simulated annealing example for a broadcast system with $M = 2, L = 2, K = 1$, and jointly Gaussian inputs. The left panel shows the initial random labelings of \mathcal{S} that served as an input to the simulated annealing algorithm. The right panel shows the final labeling. The broadcast quantizer estimate $\hat{\phi}$ is the labeled partition formed by the four regions labeled 00, 01, 10, and 11.

5. Concluding remarks

Our application of the simulated annealing algorithm requires complete knowledge of the joint input distributions under both hypotheses. In situations where the input statistics are not known or are only crudely modeled, training data could be used to estimate the cell probabilities in \mathcal{S} , if a sufficient amount of data is available.

The search complexity of this approach grows exponentially with M , L , and the resolution of \mathcal{S} . Thus, this approach is only suitable for problems with modest sized combinations of these parameters. However, simulated annealing is well-suited to incorporate a broadcast system's structural constraints whereas it is not immediately clear how to do so in a deterministic annealing or a Lagrangian formulation.

Broadcast quantizers, like other sequential quantizers, retain a significant amount of structure in their associated partitions. This structure is a direct consequent of the distributed nature of the problem and leads to low encoding complexity because each component quantizer is an easily implementable scalar quantizer. In addition, broadcast quantizers can, as demonstrated by the example, significantly boost the divergence in comparison to noncommunicative ones.

References

- [1] T. Flynn and R. M. Gray. Encoding of correlated observations. *IEEE Trans. Info. Th.*, 33(6):773–787, Nov 1987.
- [2] Allen Gersho and Robert M. Gray. *Vector Quantization and Signal Compression*. Kluwer Academic Publishers, 1992.
- [3] R. M. Gray. Quantization in task-driven sensing and distributed processing. *Proc. IEEE International Conf. on Acoustics, Speech, and Signal Processing*, pages 1049–1052, May 2006.

- [4] D. S. Johnson, C. R. Aragon, L. A. McGeoch, and C. Schevon. Optimization by simulated annealing: An experimental evaluation; Part I, Graph partitioning. *Operations Research*, 37(6):865–892, Nov-Dec 1989.
- [5] S. A. Kassam. Optimum quantization for signal detection. *IEEE Trans. Comm.*, 25(5):479–484, May 1977.
- [6] S. Kirkpatrick, C.D. Gelatt, and M.P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, May 1983.
- [7] S. Kullback. *Information Theory and Statistics*. Wiley, New York, 1959.
- [8] M. A. Lexa and D. H. Johnson. Broadcast detection structures with applications to sensor networks. *IEEE Trans. Signal Processing*. Submitted, Mar 2006.
- [9] M. Longo, T. D. Lookabaugh, and R. M. Gray. Quantization for decentralized hypothesis testing under communication constraints. *IEEE Trans. Info. Th.*, 36(2):241–255, Mar 1990.
- [10] M. Lundy and A Mees. Convergence of an annealing algorithm. *Mathematical Prog.*, 34:111–124, 1986.
- [11] H. V. Poor and J. B. Thomas. Applications of Ali-Silvey distance measures in the design of generalized quantizers for binary decision systems. *IEEE Trans. on Communications*, 25(9):893–900, Sep 1977.
- [12] D. Rebello-Monedero, R. Zhang, and B. Girod. Design of optimal quantizers for distributed source coding. In *Proc. IEEE Data Compression Conf.* IEEE Computer Society, Mar 2003.
- [13] Kenneth Rose. Deterministic annealing for clustering, compression, classification, regression, and related optimization problems. *Proc. IEEE*, 86(11):2210–2239, Nov 1998.
- [14] A. Saxena, J. Nayak, and K. Rose. On efficient quantizer design for robust distributed source coding. In *Proc. IEEE Data Compression Conf.* IEEE Computer Society, Mar 2006.
- [15] P. J. M. van Laarhoven and E. H. L. Aarts. *Simulated Annealing: Theory and Applications*. D. Reidel Publishing Co., 1987.