EE 372
Introductory Lecture
4 April 2007

Quantization, Compression, and Classification: Extracting discrete information from a continuous world

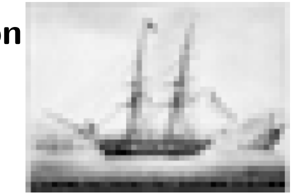
Robert M. Gray
Information Systems Laboratory, Dept of Electrical Engineering
Stanford, CA 94305
rmgray@stanford.edu

Underlying research esearch partially supported by the National Science Foundation, Norsk Electro-Optikk, and Hewlett Packard Laboratories.

Introduction

continuous world

decoder $\mathcal{D}$



$011100100\cdots$

encoder $\mathcal{E}$

discrete representations

13

decoder $\mathcal{D}$

"Picture of a privateer"

How well can you do it?
What do you mean by "well"?
How do you do it?

A dictionary definition of quantization:

— the division of a quantity into a discrete number of small parts, often assumed to be integral multiples of a common quantity.

In general, The mapping of a continuous quantity into a discrete quantity.

Converting a continuous quantity into a discrete quantity causes loss of accuracy and information.

Goal: *minimize* the loss

but need to define "loss" first . . .

and there are other constraints.

Quantization

Old example: Round off real numbers to nearest integer: estimate densities by histograms [Sheppard (1898)]

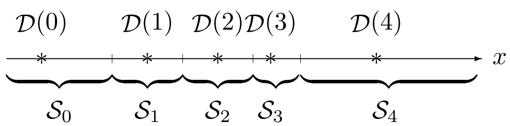
In general, quantizer q of a space A (e.g., $\Re^k$, $\mathcal{L}_2([0,1]^2)$) $=$

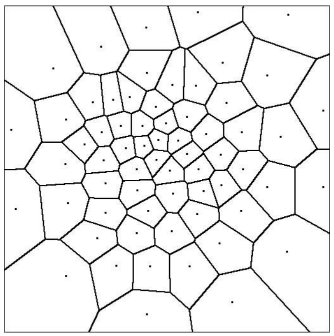
- An *encoder* $\mathcal{E} : A \to \mathcal{I}$, $\mathcal{I}$ an index set, e.g., nonnegative integers.
  $\Leftrightarrow$ *partition* $\mathcal{S} = \{S_i;\ i \in \mathcal{I}\}$: $S_i = \{x : \mathcal{E}(x) = i\}$.
- A *decoder* $\mathcal{D} : \mathcal{I} \to \mathcal{C}$ $\Leftrightarrow$ *codebook* $\mathcal{C} = \mathcal{D}(\mathcal{E}(A))$.

$$X \longrightarrow \boxed{\text{Encoder } \mathcal{E}} \xrightarrow{i = \mathcal{E}(X)} \boxed{\text{Decoder } \mathcal{D}} \longrightarrow \mathcal{D}(\mathcal{E}(X))$$

$$\boxed{\Rightarrow q = (\mathcal{E}, \mathcal{D}) = (\mathcal{S}, \mathcal{C}) = \{S_i, \mathcal{D}(i);\ i \in \mathcal{I}\}}$$

Examples Scalar quantizer:

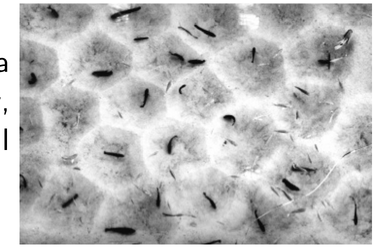


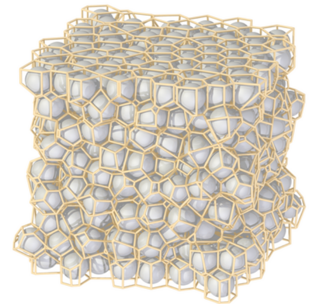


Two dimensional example: Centroidal Voronoi diagram — Nearest neighbor partition, Euclidean centroids

E.g., complex numbers, nearest mail boxes, sensors, repeaters, fish fortresses . . .

territories of the male Tilapia mossambica [G. W. Barlow, Hexagonal Territories, Animal Behavior, Volume 22, 1974]





Three dimensions: Voronoi partition around spheres http://www-math.mit.edu/dryfluids/gallery/

Performance: Distortion

Quality of a quantizer measured by the goodness of the resulting reproduction in comparison to the original.

Assume a *distortion measure* $d(x, y)$ which measures penalty/cost if an input x results in an output $y = \mathcal{D}(\mathcal{E}(x))$.

Small (large) distortion $\Leftrightarrow$ good (bad) quality

Performance measured by *average distortion*: assume X is a random object (variable, vector, process, field) described by probability distribution P_X: pixel intensities, samples, features, fields, DCT or wavelet transform coefficients, moments, . . .

E.g., if $A = \Re^k$, $P_X \Leftrightarrow$ pdf f, or empirical distribution $P_{\mathcal{L}}$ from training/learning set $\mathcal{L} = \{x_l;\ l = 1, 2, \ldots, |\mathcal{L}|\}$

Average Distortion:

$$\begin{aligned} D(q) = D(\mathcal{E}, \mathcal{D}) &= E[d(X, \mathcal{D}(\mathcal{E}(X))] \\ &= \int d(x, \mathcal{D}(\mathcal{E}(x))\, dP_X(x) \end{aligned}$$

Examples: $x, y \in \Re^k$

- Mean squared error (MSE) $d(x, y) = ||x - y||^2 = \sum_{i=1}^{k} |x_i - y_i|^2$

- Input or output weighted quadratic, B_x positive definite matrix: $d(x, y) = (x - y)^t B_x (x - y)$ or $(x - y)^t B_y (x - y)$

Used in perceptual coding, statistical classification.

- Functions of norms:

$$d(x,y) = \rho(||x-y||_p)$$

where ℓ_p norm for $p \geq 1$ is

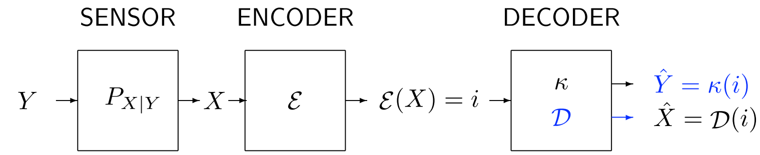
$$||x-y||_p = \left(\sum_{i=0}^{k-1} |x_i - y_i|^p \right)^{1/p}$$

ρ is convex.

Mostly of mathematical interest, limited demonstrated usefulness in applications.

- Vector X might be pmf, distortion could be distribution distance, relative entropy (Kullback-Leibler distance), etc.

- Bayes distortion, "task-driven" quantization.



e.g., $X = Y+$ noise remote or noisy source coding

"denoise" by compression

Overall distortion: Bayes risk $C(y, \hat{y})$ Define quantizer distortion

$$\begin{aligned} d(x, \kappa(i)) &= E[C(Y, \kappa(i))|X = x] \Rightarrow \\ E[d(X, \kappa(\mathcal{E}(X))] &= E[C(Y, \kappa(\mathcal{E}(X)))] \end{aligned}$$

Y discrete $\Rightarrow$ classification, detection
Y continuous $\Rightarrow$ estimation, regression

Performance: Rate

Want average distortion small, doable with big codebooks and small cells, **but** there is usually a *cost* (*instantaneous rate*) $r(i)$ of choosing index i, with average rate $R(q) = E[r(\mathcal{E}(X))]$, e.g.,

- $r(i) = \ln |\mathcal{C}|$ Fixed-rate, classic quantization
- $r(i) = \ell(i)$ Variable-rate quantization, *length function* ℓ satisfies Kraft inequality $\sum_i e^{-\ell(i)} \leq 1$
- $r(i) = -\ln p(i)$, $p(i) = \Pr(\mathcal{E}(X) = i)$ Shannon codelengths
  $R(q) = H(\mathcal{E}(X)) = -\sum_i p(i) \ln p(i)$ entropy coding
- $r(i) = (1-\eta)\ell(i) + \eta \ln |\mathcal{C}|$, $\eta \in [0,1]$ Combined constraints
  Proposed by Zador (1982) to unify separate cases.

Quantizer Model

ℓ best thought of as part of quantizer, a design choice.

Alternative view: Codeword weighting: $w(i) = e^{-\ell(i)}$, $w \Leftrightarrow \ell$

$\ell(i) = \infty \Leftrightarrow w(i) = 0$; infinite cost, never used.

$N(w) = |\{i : w(i) > 0\}| = |\{i : \ell(i) \text{ finite }\}|$ codebook size

Kraft inequality on $\ell \Leftrightarrow w$ a sub-pmf: $w(i) \geq 0$, $\sum_i w(i) \leq 1$

$r(i) = (1-\eta) \ln \frac{1}{w(i)} + \eta \ln N(w)$,

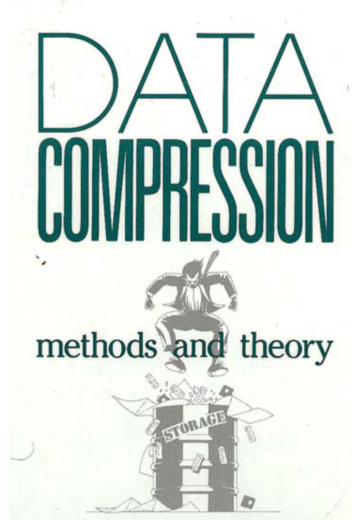
$\Rightarrow p$ must be absolutely continuous wrt w for finite rate.

quantizer model is $\boxed{q = (\mathcal{E}, \mathcal{D}, \ell) = (\mathcal{S}, \mathcal{C}, w)}$

Major Issues

- If *know* distributions,
  - what is *optimal* tradeoff of distortion $D(\mathcal{E}, \mathcal{D})$ vs. rate $R(\mathcal{E}, w)$? (Want *both* to be small!)
  - How design good codes?
- If do *not* know distributions, how use training/learning data to estimate tradeoffs and design good codes?
  (clustering, statistical or machine learning)

Applications

- 

- Communications & signal processing: A/D conversion, data compression. Compression required for efficient **transmission**
  - send more data in available bandwidth
  - send the same data in less bandwidth
  - more users on same bandwidth

  and **storage**

Graphic courtesy of Jim Storer.

- Statistical clustering: grouping bird songs, designing Mao suits, grouping gene features, taxonomy.
- Placing points in space: mailboxes, wireless sensors
- Image and video classification/segmentation
- Speech recognition and speaker identification
- Numerical integration and optimal quadrature rules, e.g., How best chose $\{S_i, \mathcal{D}(i)\}$ for $\int g(x) f(x)\, dx \approx \sum_i P_X(S_i) g(\mathcal{D}(i))$?
- Approximating continuous probabilistic models by discrete models. Simulating continuous processes.
- Fitting and choosing Gauss mixture models for observed data.

Distortion-rate optimization: $q = (\mathcal{E}, \mathcal{D}, w)$

$$D(q) = E[d(X, \mathcal{D}(\mathcal{E}(X)))] \text{ vs. } R(q) = E[r(\mathcal{E}(X))]$$

Minimize $D(q)$ for $R(q) \leq R$ $\qquad \delta(R) = \inf_{q: R(q) \leq R} D(q)$

Minimize $R(q)$ for $D(q) \leq D$ $\qquad r(D) = \inf_{q: D(q) \leq D} R(q)$

Lagrangian approach: $\rho_\lambda(x, i) = d(x, \mathcal{D}(i)) + \lambda r(i)$, $\lambda \geq 0$

Minimize Lagrangian distortion $\rho(f, \lambda, \eta, q) \triangleq E\rho_\lambda(X, \mathcal{E}(X))$

$$= D(q) + \lambda R(q)$$

$$= Ed(X, \mathcal{D}(\mathcal{E}(X))) + \lambda \left[(1-\eta) E\ell(\mathcal{E}(X)) + \eta \ln N(\ell)\right]$$

$$\rho(f, \lambda, \eta) = \inf_q \rho(f, \lambda, \eta, q)$$

Traditional cases: fixed-rate $\eta = 1$, variable-rate $\eta = 0$

Three Theories of Quantization

Rate-Distortion Theory Shannon distortion-rate function $D(R) \leq \delta(R)$, achievable in asymptopia of $\boxed{\text{large dimension } k}$ and fixed rate R. [Shannon (1949, 1959), Gallager (1978)]

⋆ **Nonasymptotic (Exact) results** Necessary conditions for optimal codes ⇒ iterative design algorithms
⇔ statistical clustering [Steinhaus (1956), Lloyd (1957)]

⋆ **High rate theory** Optimal performance in asymptopia of fixed dimension k and $\boxed{\text{large rate } R}$. [Bennett (1948), Lloyd (1957), Zador (1963), Gersho (1979), Bucklew, Wise (1982)]

Lloyd Optimality Properties: $q = (\mathcal{E}, \mathcal{D}, \ell)$

Any component can be optimized for the others.

Encoder	$\mathcal{E}(x) = \operatorname*{argmin}_i \left(d(x, \mathcal{D}(i)) + \lambda r(i)\right)$	Minimum distortion
Decoder	$\mathcal{D}(i) = \operatorname*{argmin}_y E(d(X,y) \mid \mathcal{E}(X) = i)$	Lloyd centroid
Codebook size	Remove i if $\Pr(\mathcal{E}(X) = i) = 0$	Prune
Length function	$\ell(i) = -\ln P_X(\mathcal{E}(X) = i)$	Shannon codelength

Iterate steps ⇒ Lloyd *clustering* algorithm, (rediscovered as *k-means*, *grouped coordinate descent*, *alternating optimization*, *principal points*)

If fixed length, squared error $\Longrightarrow$ *Centroidal Voronoi partition*

centroid $= \mathcal{D}(i) = E(X \mid \mathcal{E}(X) = i)$ (MMSE estimate)

Lloyd conditions ⇒

- Encoder partition $\mathcal{S}$ ⇒ optimal decoder $\mathcal{D} = \mathcal{D}(\mathcal{S})$ and length function $\ell = \ell(\mathcal{S})$
- Decoder $\mathcal{D}$ + length function ℓ ⇒ optimal encoder partition $\mathcal{S} = \mathcal{S}(\mathcal{D}, \ell)$

Equivalent optimization problems:

$$\begin{aligned} \rho(f, \lambda, \eta) &= \inf_{\mathcal{S}, \mathcal{D}, w} \rho(f, \lambda, \eta, \mathcal{S}, \mathcal{D}, w) \\ &= \inf_{\mathcal{S}} \rho(f, \lambda, \eta, \mathcal{S}) \\ &= \inf_{\mathcal{D}, w} \rho(f, \lambda, \eta, \mathcal{D}, w) \end{aligned}$$

Good quantizer characterized by either partition $\mathcal{S}$ or by weighted decoder $(\mathcal{D}, w)$. Suggests another Lloyd-style optimality condition.

Subcodes and Supercodes, Pruning and Growing

A partition $\mathcal{S}'$ is a subpartition of a partition $\mathcal{S}$ if every atom of $\mathcal{S}'$ is a union of atoms of $\mathcal{S}$: $\mathcal{S}$ *refines* $\mathcal{S}'$ or $\mathcal{S}' \subset \mathcal{S}$

A quantizer q' determined by $\mathcal{S}'$ is a *partition subcode* of the quantizer q determined by $\mathcal{S}$ if $\mathcal{S}'$ is a subpartition of $\mathcal{S}$. q is a *partition supercode* of q'.

A weighted codebook $(\mathcal{D}', w')$ is a *codebook subcode* of a weighted codebook $(\mathcal{D}, w)$ if $\{\mathcal{D}'(i), i : w'(i) > 0\} \subset \{\mathcal{D}(i), i : w(i) > 0\}$ and $w'_i = \alpha w_i, \alpha \in (0,1)$. Conversely, $(\mathcal{D}, w)$ is a codebook supercode of $(\mathcal{D}', w')$.

Generally the smaller code has larger distortion and smaller rate.

Provides an additional Lloyd-style necessary condition:

Codebook size If a quantizer q is optimal there can be no subcode or supercode q' for which $D(q') + \lambda R(q') < D(q) + \lambda R(q)$. pruning/growing

Generalization of zero-probability cell pruning.

Can test likely subcodes/supercodes for possible improvement.

Can also use to increase/decrease λ and grow or prune code.

Combining everything $\Rightarrow$ Lloyd clustering algorithm

A descent algorithm, so distortion converges.

Step 0: Initialization Given initial $(\mathcal{D}_0, w_0)$. Compute $\rho_0 = \rho(\mathcal{S}(\mathcal{D}_0, w_0), \mathcal{D}_0, w_0)$. Set $m = 1$

Step 1: Partition improvement Given$(\mathcal{D}_{m-1}, w_{m-1})$, form an optimum partition $\mathcal{S}_m = \mathcal{S}(\mathcal{D}_{m-1}, w_{m-1})$.

Step 2: Weighted codebook improvement Given the partition $\mathcal{S}_m$, form an optimum weighted codebook $(\mathcal{D}_m, w_m) = (\mathcal{D}(\mathcal{S}_m), w(\mathcal{S}_m))$. Compute $\rho_m = \rho(\mathcal{S}_m, \mathcal{D}_m, w_m)$.

Step 3: Test Test $\rho_{m-1} - \rho_m$. If small enough, go to Step 4. Else set $m = m + 1$, go to Step 1.

Step 4: Grow/Prune Test sub/super codes for possible improvement for fixed λ or for changed λ. Quit or go to step 1.

Bayes quantization

If use Bayes distortion for statistical classification or regression with a rate constraint, e.g., with Hamming Bayes cost: $E[C(Y, \kappa(\mathcal{E}(X)))] = P_e$ and

$$\begin{aligned}
\kappa(i) &= \underset{\hat{y}}{\operatorname{argmax}} \Pr(Y = \hat{y} | \mathcal{E}(X) = i) \\
\mathcal{E}(x) &= \underset{i}{\operatorname{argmax}} [\Pr(Y = \kappa(i) | X = x) - \lambda r(i)]
\end{aligned}$$

Need to estimate $P_{Y|X}$ via model or binning/quantization of X

Can add constraints, e.g., constrain encoder to have low-complexity tree structure. (CART/BFOS/TSVQ)

High Rate (Resolution) Theory

Traditional form (Zador, Gersho, Bucklew, Wise): f, MSE

$$\lim_{R \to \infty} e^{\frac{2}{k}R} \delta(R) = \begin{cases} a_k ||f||_{\frac{k}{k+2}} & \text{fixed-rate } (R = \ln N) \\ b_k e^{\frac{2}{k}h(f)} & \text{variable-rate } (R = H)\text{, where} \end{cases}$$

$$h(f) = -\int f(x) \ln f(x)\, dx, \quad ||f||_{k/(k+2)} = \left(\int f(x)^{\frac{k}{k+2}}\, dx \right)^{\frac{k+2}{k}}$$

$a_k \geq b_k$ are Zador constants (depend on k and d, not f!)

$$a_1 = b_1 = \frac{1}{12}, a_2 = \frac{5}{18\sqrt{3}}, b_2 = ?, a_k, b_k = ? \text{ for } k \geq 3$$

$$\lim_{k \to \infty} a_k = \lim_{k \to \infty} b_k = 1/2\pi e$$

Lagrangian form: variable-rate

$$\begin{aligned}\lim_{\lambda\to 0}\left[\inf_q\left(\frac{\rho(f,\lambda,0,q)}{\lambda}\right)+\frac{k}{2}\ln\lambda\right] &= \theta_k + h(f)\\ \theta_k \triangleq \inf_{\lambda>0}\left[\inf_q\left(\frac{\rho(u,\lambda,0,q)}{\lambda}\right)+\frac{k}{2}\ln\lambda\right] &= \frac{k}{2}\ln\frac{2eb_k}{k}\end{aligned}$$

$$\begin{aligned}\lim_{\lambda\to 0}\left[\inf_q\left(\frac{\rho(f,\lambda,1,q)}{\lambda}\right)+\frac{k}{2}\ln\lambda\right] &= \psi_k + \ln ||f||_{k/(k+2)}^{k/2}\\ \psi_k \triangleq \inf_{\lambda>0}\left[\inf_q\left(\frac{\rho(u,\lambda,1,q)}{\lambda}\right)+\frac{k}{2}\ln\lambda\right] &= \frac{k}{2}\ln\frac{2ea_k}{k}\end{aligned}$$

fixed-rate both have form optimum for uniform u + function of f

Proofs

Rigorous proofs of traditional cases painfully tedious, but follow Zador's original approach.

Heuristic proofs developed by Gersho based on assumptions of

- existence of asymptotically optimal quantizer point density function $\Lambda(x)$
- existence of asymptotically optimal quantizer partition cell shape

Then hand-waving approximations of integrals relate $N(q), D_f(q), H_f(q) \Rightarrow$

$$\begin{aligned}D_f(q) &\approx c_k E_f\left(\left(\frac{1}{N(q)\Lambda(X)}\right)^{2/k}\right)\\ H_f(q(X)) &\approx h(X) - E\left(\log\left(\frac{1}{N(q)\Lambda(X)}\right)\right)\\ &= \ln N(q) - \underbrace{\int f(x)\ln\frac{f(x)}{\Lambda(x)}\,dx}_{H(f||\Lambda)\text{ relative entropy}},\end{aligned}$$

Optimizing using Holder's inequality or Jensen's inequality yields classic fixed-rate and variable-rate results.

Can also apply to combined-constraint case to get conjectured solution for general η

There appear to be connections with rigorous and heuristic proofs, may help insight and simplify proofs.

Gersho's conjectures and approximations have many yet unproved but commonly believed implications,

e.g., $a_k = b_k$ all k, the best fixed-rate high rate codes have maximum entropy, asymptotically optimal quantizer point density functions exist.

Henceforth focus on variable-rate case, $\eta = 0$.

Mismatch distortion

High-rate results ⇒ for small λ optimal quantizer satisfies

$$\boxed{\frac{\rho(f,\lambda,0,q)}{\lambda}+\frac{k}{2}\ln\lambda \approx \theta_k + h(f)}$$

Worst case: Performance depends on f only through $h(f)$ ⇒ worst case source given constraints is max h source, e.g., given m and K, worst case is Gaussian

Mismatch: Design for pdf g, but apply it to f for small λ

$$\boxed{\frac{\rho(f,\lambda,0,q)}{\lambda}+\frac{k}{2}\ln\lambda \approx \underbrace{h(f)+\theta_k}_{\text{optimal for } f} + \underbrace{H(f\|g)}_{\text{mismatch}}\ .}$$

Put together: If g Gaussian with same mean, covariance as f then

$$\frac{\rho(f,\lambda,0,q)}{\lambda}-\frac{\rho(g,\lambda,0,q)}{\lambda}\approx 0$$

⇒ the performance of the code designed for g asymptotically equals that when the code is applied to f, a *robust* code in information theoretic sense!
(≈ "generalization guarantees" in machine learning)

Robust and worst-case code ⇒ $H(f\|g)$ measures *mismatch distance* from f to Gaussian g with same second-order moments, I.e., for a pdf f, how much performance is lost from optimal for f if use optimal code for g on f?

Problem with single worst case: too conservative

Gaussian can be a very bad fit, very suboptimal, e.g., speech.

Divide & conquer: composite code

Alternative idea, fit a *different* Gauss source to distinct groups of inputs instead of the entire collection of inputs. Robust codes for local behavior.

Partition input space $\Re^k$ into $\mathcal{S}=\{S_m;\ m=1,\ldots,M\}$, assign a Gaussian component $g_m=\mathcal{N}(\mu_m,K_m)$ to each S_m, & construct an optimal quantizer $q_m=(\mathcal{E}_m,\mathcal{D}_m,\ell_m)$ for each g_m:

$$\frac{\rho(f,\lambda,0,q)}{\lambda}+\frac{k}{2}\ln\lambda \approx \underbrace{\theta_k+h(f)}_{\text{optimal for } f} + \underbrace{\sum_m p_m H(f_m\|g_m)}_{\text{mismatch distortion}}$$

where $f_m(x)=f(x)/p_m$, $x\in S_m$, $p_m=\int_{S_m} dx\, f(x)$.

What is best (smallest) can make $\sum_m p_m H(f_m\|g_m)$ over all partitions $\mathcal{S}=\{S_m;\ m\in\mathcal{I}\}$ and collections $\mathcal{G}=\{g_m,p_m;\ m\in\mathcal{I}\}$?

$$\Leftrightarrow$$

Quantize $\Re^k$ into space of Gauss models using quantizer mismatch distortion

$$d_{\mathrm{QM}}(x,m)=-\ln p_m+\frac{1}{2}\ln|K_m|+\frac{1}{2}(x-\mu_m)^t K_m^{-1}(x-\mu_m)$$

Similar to MDI distortion, Itakura-Saito distortion, log likelihood distortion for Gaussians, Stein's spectral distortion, . . .

Codebook $\{(\mu_m,K_m,p_m)\}\leftrightarrow$ Gauss mixture model.

Can use Lloyd algorithm to optimize: *Gauss mixture vector quantization (GMVQ)*

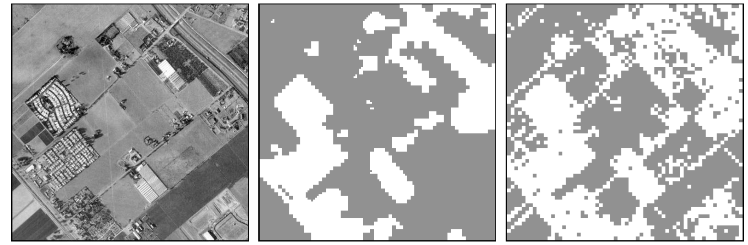
- GMVQ converts a training sequence into a GM model
- Derivation was based on high-rate analysis of quantizing the original source using a collection of codes designed for local Gaussian models. Minimizing the overall MSE $\Leftrightarrow$ GMVQ
- Alternative to Baum-Welch/EM algorithm for GM design.

Advantages over EM:

- Lower complexity (about 1/2).
- Faster convergence.
- Incorporates compression $\Rightarrow$ can use low complexity search, e.g., TSVQ, useful for large dimensions, large datasets, networks.

Examples

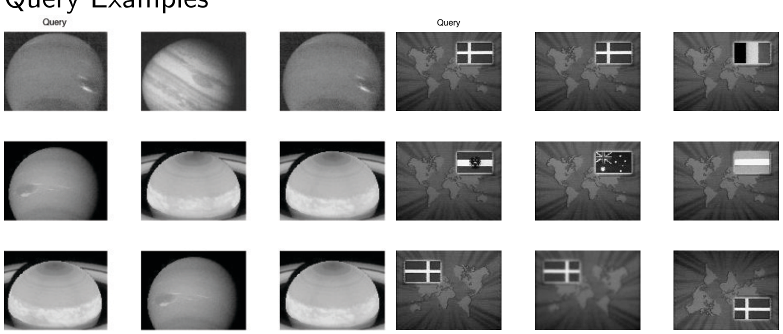
Aerial images: Manmade vs. natural White: man-made, gray:natural



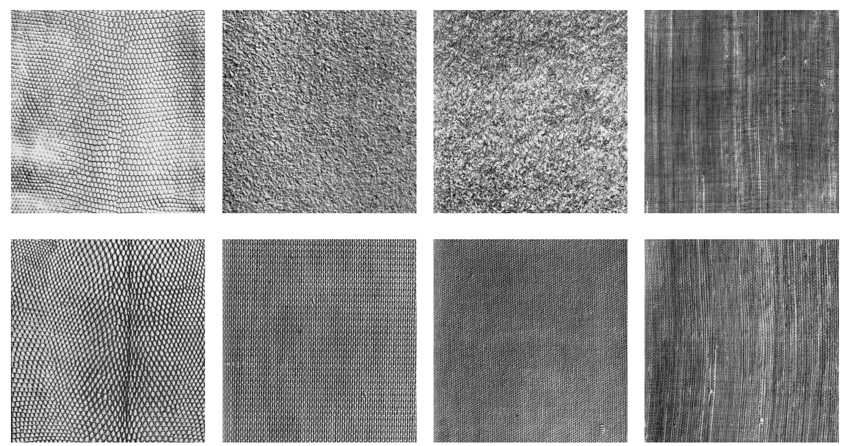
original 8 bpp, gray-scale images, hand-labeled classified images. GMVQ with probability of error 12.23%

Content-Adresssible Databases/Image Retrieval

8×8 blocks, 9 images train for each of 50 classes, 500 image database (cross-validated)
Precision (fraction of retrieved images that are relevant) $\approx$
recall (fraction of relevant images that are retrieved) $\approx .94$.
Query Examples

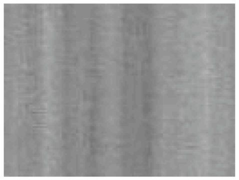


Texture Classification

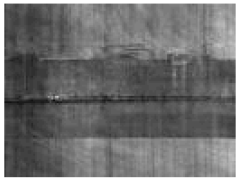


Examples from Brodatz database
Compared well with Gauss mixture random field methods.

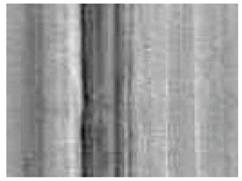
North Sea Gas Pipeline Data



normal pipeline image



field joint



longitudinal weld

which is which???

- GMVQ Best codebook wins
- MAP-GMVQ Plug in GMVQ produced GMM to MAP
- MAP-EM: Plug into EM produced GMM to MAP
- 1-NN
- MART (boosted classification tree)
- Regularized QDA (Gaussian class models)
- MAP-ECVQ (GM class models)

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**
GMVQ	0.9673	0.8000	0.9487	0.9737	0.7619	0.9487	**0.9481**

Class	**Subclasses**
S	Normal, Osmosis Blisters, Black Lines, Small Black Corrosion Dots, Grinder Marks, MFL Marks, Corrosion Blisters, Single Dots
V	Longitudinal Welds
W	Weld Cavity, Field Joint

Recall $= \Pr$ (declare as class i | class i)
Precision $=\Pr$ (class i | declare as class i)

Implementation is simple, convergence is fast, classification performance is good.

Closing Introductory Thoughts

- Quantization ideas useful for A/D conversion, quantizer error analysis, data compression, statistical classification, modeling, and estimation with a rate constraint.
- Lloyd-designed GMVQ alternative to EM for Gauss mixture design in statistical classification and clustering.
- Gersho's approach provides intuitive, but nonrigorous, proofs of high rate results based on conjectured geometry of optimal cell shapes.

  New tools may help develop rigorous proof reflecting simple heuristics & means of proving/disproving implications of Gersho's conjecture.
- Basic tools include information theory, signal processing, optimization.