Flexible Quantization

Bastiaan Kleijn
KTH School of Electrical Engineering
Stockholm

Overview

- Motivation for coding technologies
- Basic quantization and coding
- High-rate quantization theory

Digital Representations

- Digital representation of signal
  - Sequence of samples with finite precision
  - Robust against distortion
  - Facilitates processing

- Basic rates for audio signals:
  - 48 kHz audio, 16 bits, stereo: 1536000 bits/second
  - 8 kHz speech, 16 bits: 128000 bits/second

High Rate is Expensive

- Transmission
  - Wired links
    - "Last mile"
    - Packet networks
    - Switching video
  - Wireless links
    - WiFi
    - Mobile telephony: coding was enabling technology
  - Secure communication
- Storage
  - Portable audio/video players
  - Output surveillance cameras

High Quality Now Less Natural

- Conventional circuit-switched networks
  - Virtually no bit errors, no loss
- Mobile networks
  - Reasonable cost and delay implies bit errors
- Packet networks
  - Reasonable cost and delay implies packet loss

Networks More Diverse

- How it was:
  - Single-paradigm network end-to-end
  - One service
- How it is:
  - Many paradigms in one composite network:
    - Circuit-switched network
    - Packet network
    - Wireless circuit-switched network
    - Wireless packet network
  - Many types of service
    - Range of quality-cost
    - Streaming, one-on-one communication

How We Designed Coders

- New application (particular network, storage) appeared
- Study application requirements
- Design coder for application requirements
- Have competition between coder designs for conditions of application
- Select best coder

- No vision of integrated network

More On Design Conditions

- Attributes of a coder
  – Rate
  – Quality (subjective), includes signal bandwidth
  – Delay
  – Robustness: bit errors and packet loss
  – Computational complexity

- Designs selected for one configuration of attributes
  – Associated with one network paradigm
  – Design effort irrelevant

Adapting to the New Environment

- Implications of old-school design in new world:
  - Coders implicitly unable to adapt: codebooks
  - Transcoding
  - Performance unclear when applied to other conditions
    - GSM coder applied to packet networks

- New-school design
  - Goal: coders that can adapt in real-time to
    - Network conditions
    - Quality requirements
  - Near-optimal over large range of conditions
  - Employ high-rate quantization theory and more modeling

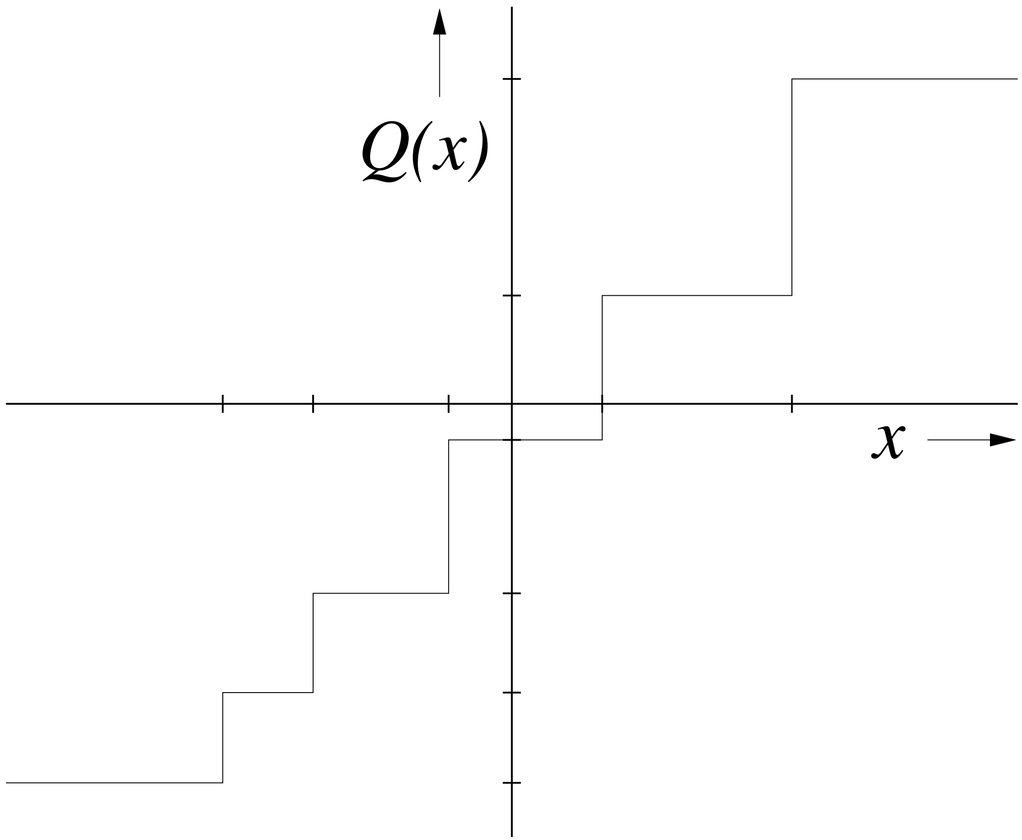
Overview

- Motivation for coding technologies
- Basic quantization and coding
- High-rate quantization theory

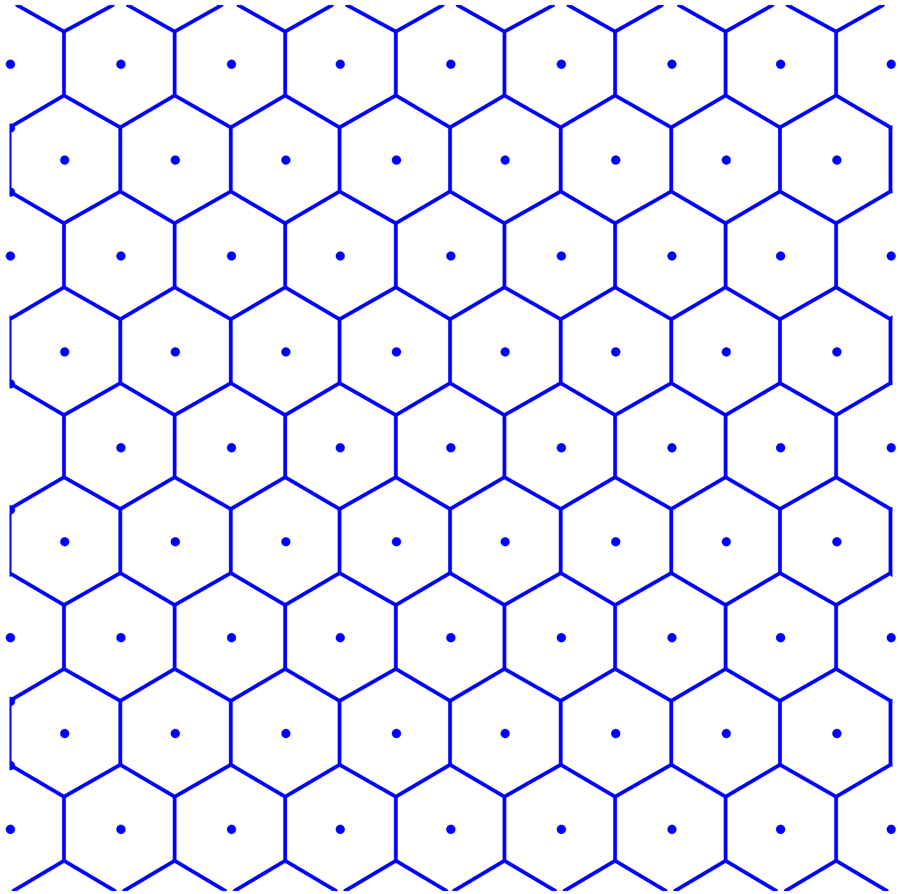
Quantization

- Quantization: non-invertible mapping from Euclidian space $\mathscr{R}^k$ to a countable set of points $\mathcal{C} = \{c_i\}$ that is a subset of $\mathscr{R}^k$
  - Quantization cell: $V_i = \{x \in \mathscr{R}^k : Q(x) = c_i\}$
  - "Inverse quantization" is misnomer

Example: Scalar Quantizer



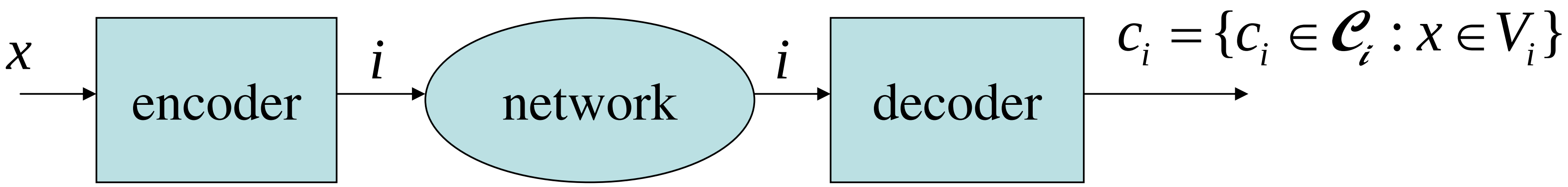
Example: Vector Quantizer



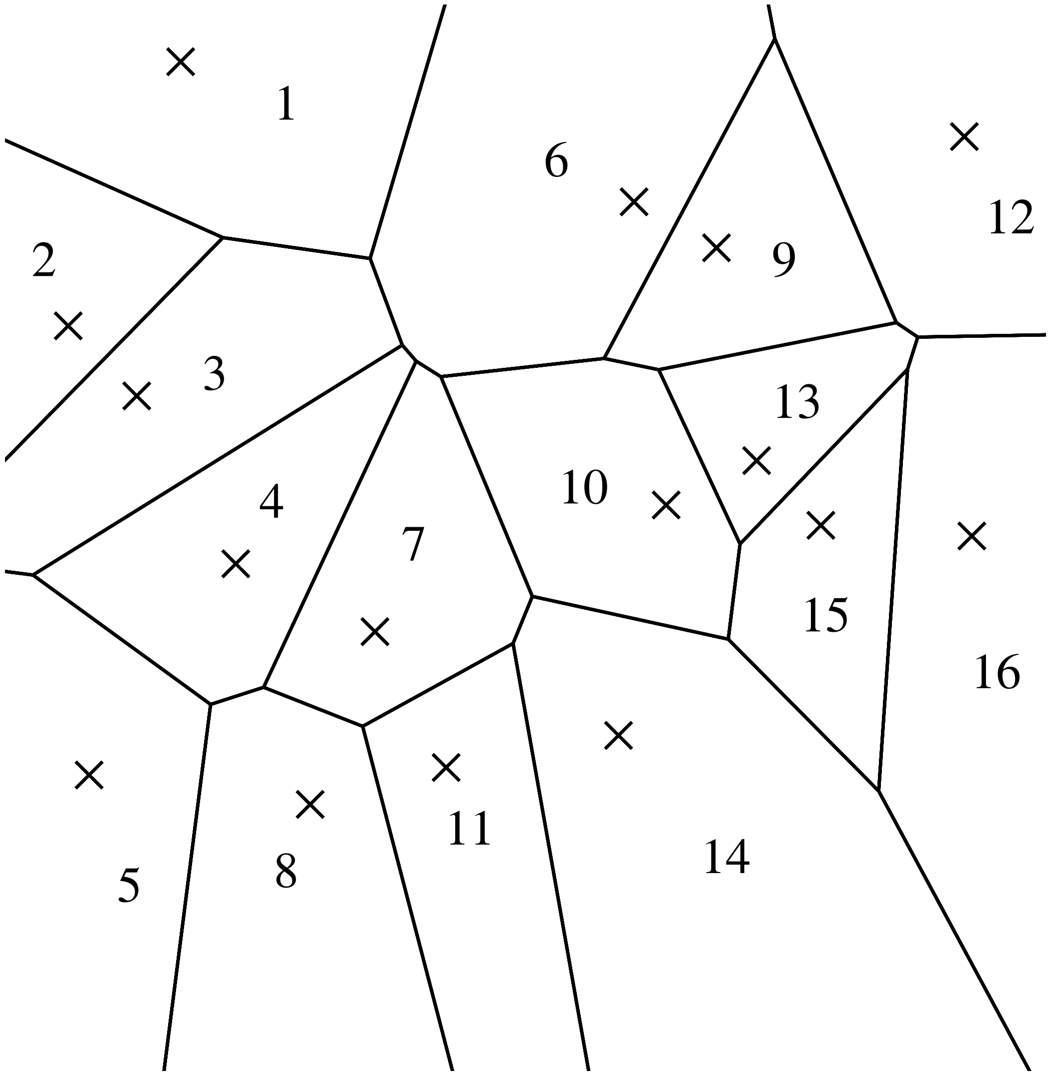
Quantization Cells and Centroids

- $V_i = \{x \in \mathcal{R}^k : Q(x) = c_i\}$ is a cell
  - Usually assumed convex: "*regular*" quantizers
  - Cell = Voronoi region
- The quantization index i specifies the cell and the reconstruction point (often called the centroid)

- If the set of indices $\{i\}$ is countable, the quantization index can generally be transmitted with a finite number of bits



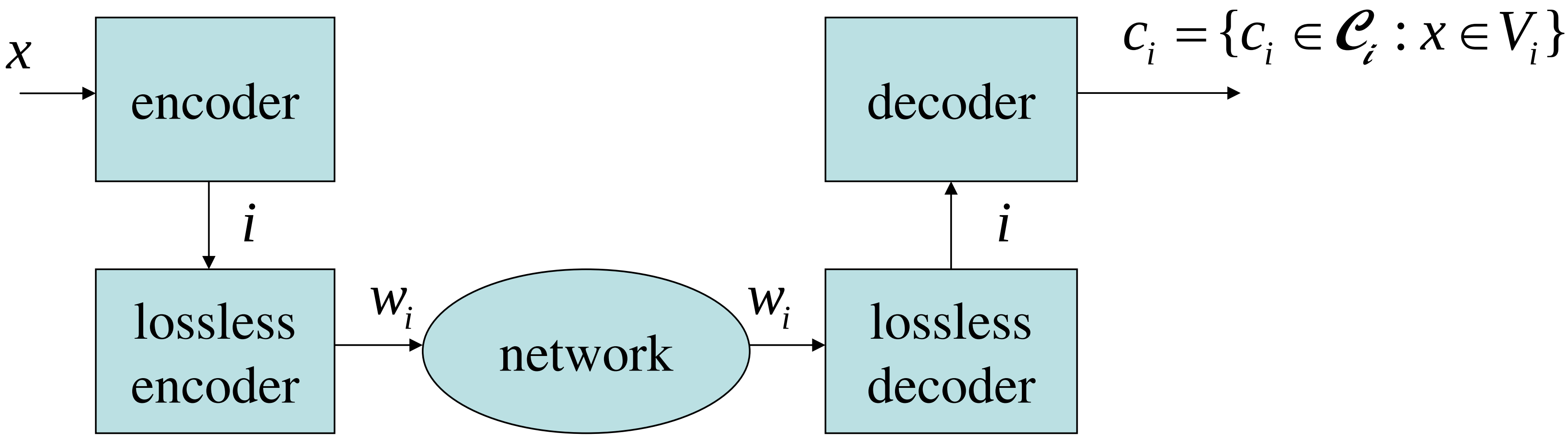
Example: Vector Quantizer



Coding Principles

- Is it smart to simply transmit the index i? NO!
  - (it is if index probability uniform)

- Apply lossless (entropy) coding to indices
  - Used to create ".zip"



Minimum (Bit) Rate of Index

- Code: the set of all codewords $\{w_i\}$
- Uniquely decodable code: can always reconstruct
- ~Minimum codeword length for uniquely decodable code:
  $l(w_i) = -\log_2(p_I(i))$ (follows from *Kraft inequality*)
- Entropy of the index: $\boxed{H(I) = -\sum_i p_I(i)\,\log_2(p_I(i))}$
  - Is ~minimum average rate needed for index
    - More accurately: $H(I) \le L < H(I)+1$

Example

- Index resembles coin flips

$$H(I) = -\sum_i p_I(i)\,\log_2(p_I(i)) = -0.5\,\log_2(0.5) - 0.5\,\log_2(0.5) = 1\ \text{bits}$$

- Index resembles biased coin flips

$$\begin{aligned} H(I) &= -0.25\,\log_2(0.25) - 0.75\,\log_2(0.75) = 0.811\ \text{bits} \\ &= -0.05\,\log_2(0.05) - 0.95\,\log_2(0.95) = 0.286\ \text{bits} \end{aligned}$$

Now Back to Quantization

- To quantize we need to know with respect to what

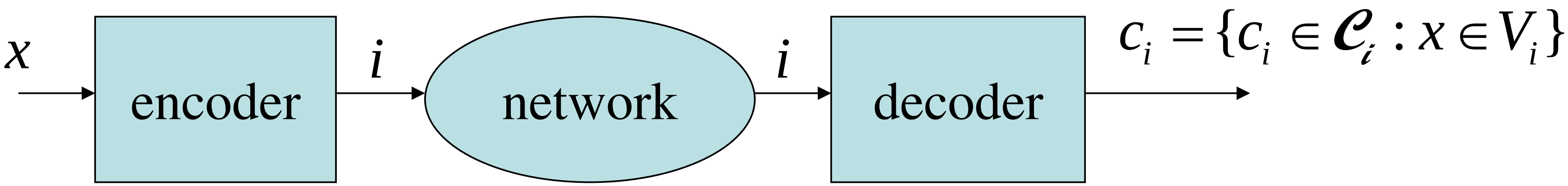
- Optimal trade-off *distortion versus number of indices*
  - *Constrained-resolution*
  - Assumes codeword length is fixed
  - Generally short delay
  - Consistent with TDMA and FDMA, circuit-switched networks
  - The past

- Optimal trade-off *distortion versus average rate*
  - *Constrained-entropy*
  - Assumes only average codeword length matters
  - Often long delay
  - Consistent with CDMA and packet-switched networks
  - The future!

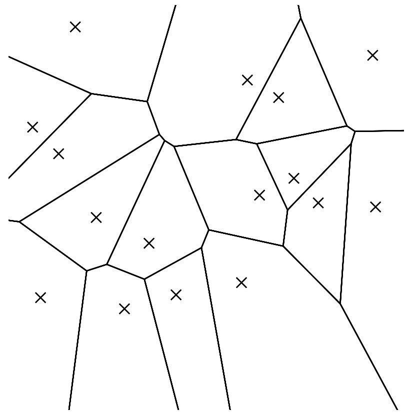
Old-School, Any-Rate Quantization

- Standard approach
  - Constrained-resolution
  - Stored codebooks
  - Codebooks trained with data
    - (Generalized) Lloyd algorithm (GLA), Bell Labs, 1958 / K-means algorithm

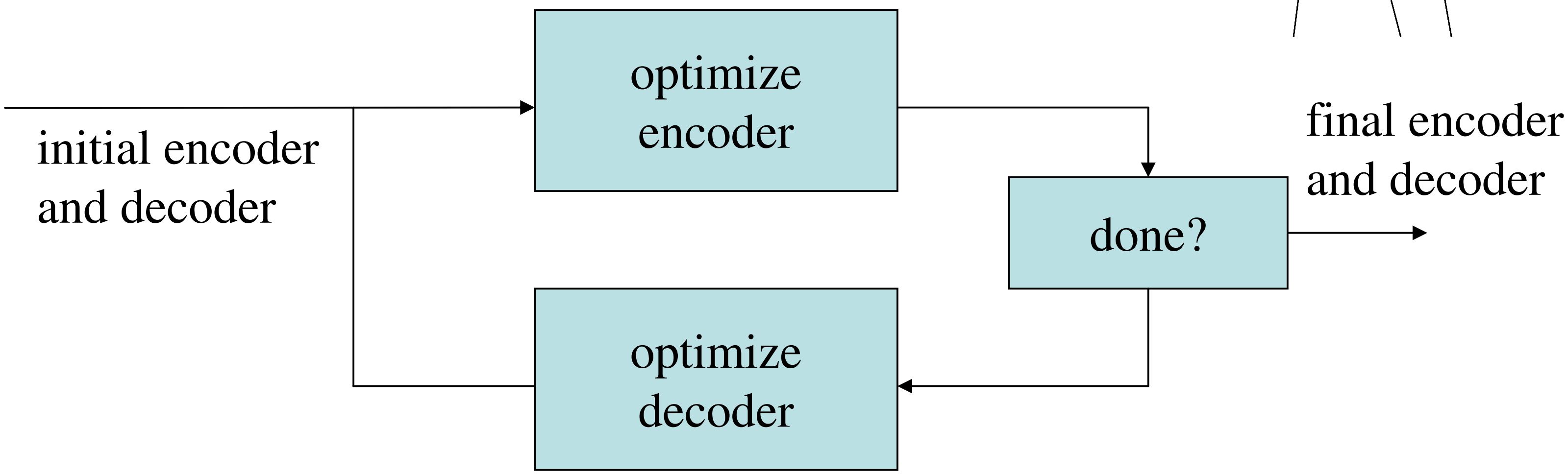
- Is constrained-resolution



Lloyd Algorithm

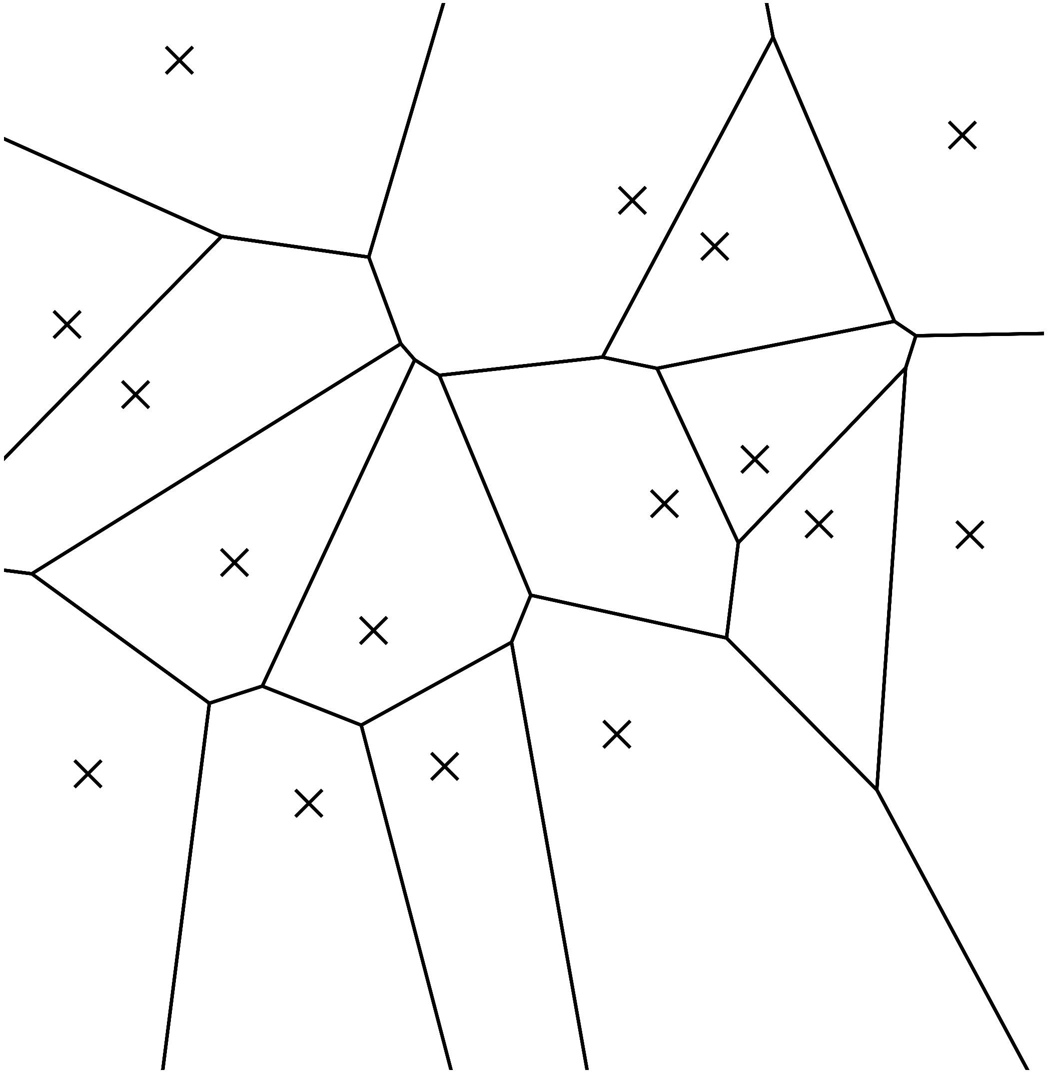


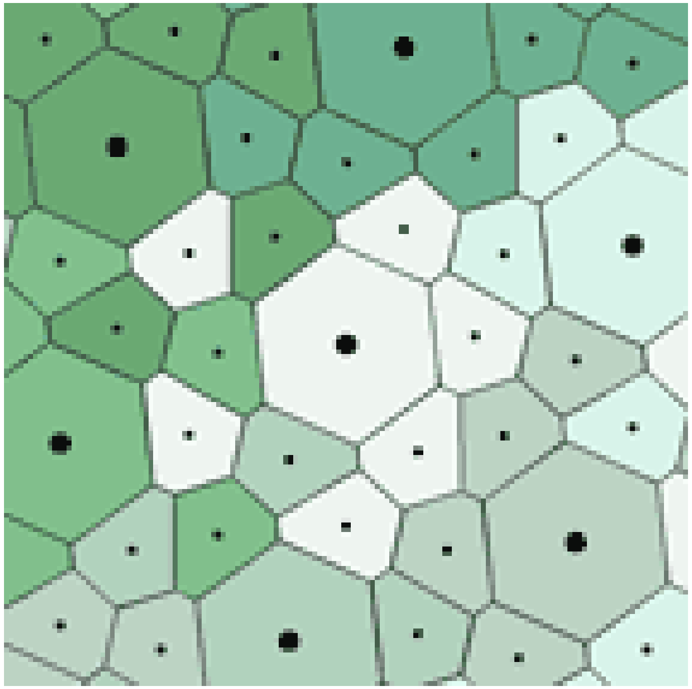
- Note:
  - Encoder = Partition={Voronoi regions}
  - Decoder = codebook={centroids)
- Lloyd algorithm:



- Optimize: minimize mean distortion: $\mathrm{E}[\min_{i\in I} d(X^k, c_i)]$
- Locally optimal

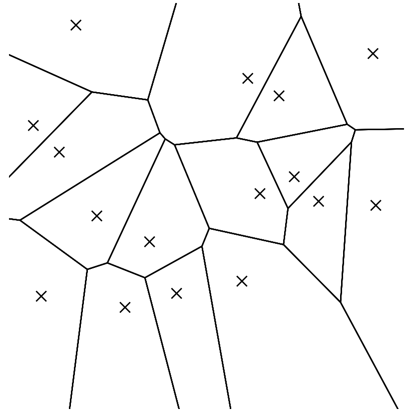
Outcome Lloyd for Vector Quantizer

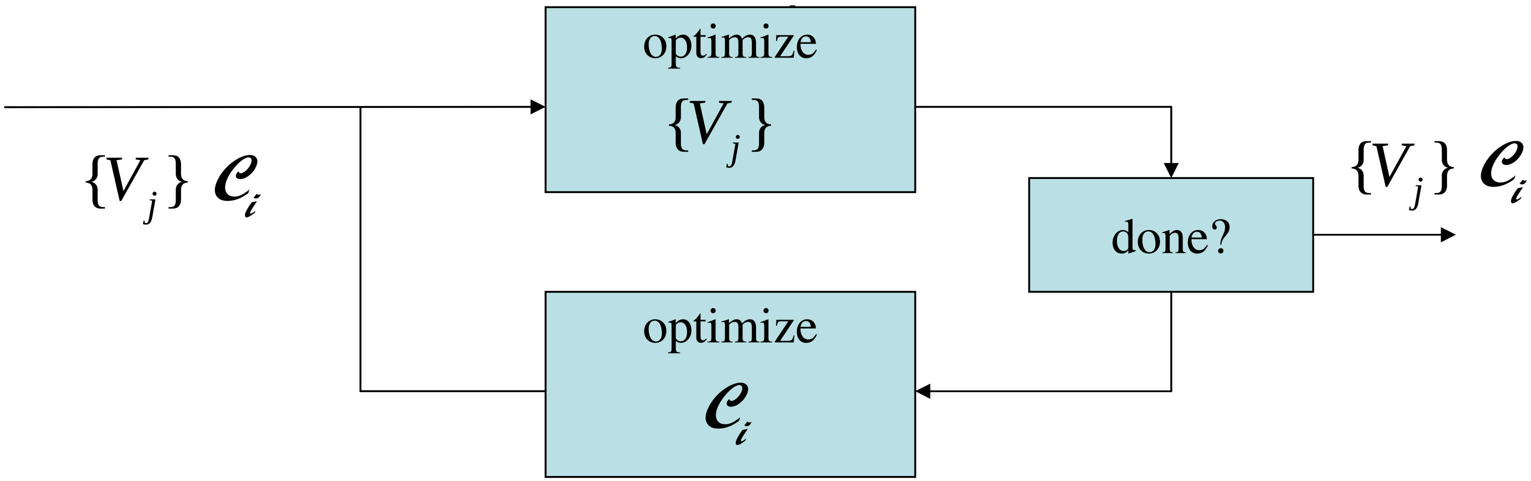




Practical (Discrete) Lloyd Algorithm

- Have database $\{x_m^k\}_{m\in M}$
- Encoder = partition = $\{V_j = \{x_i^k\}_{i\in \mathcal{J}(j)} : \bigcup V_j = \{x_i^k\}_{i\in I}\}$
- Decoder = codebook = $\mathcal{C}_i = \{c_i\}$





- Optimize = minimize overall distortion: $\sum_{m\in M}[\min_{i\in I} d(x_m^k,\ c_i)]$

Old-School, Any-Rate Quantization

- Is in your cell phone
- Constrained resolution (fixed number of cells/centroids)
- Works even at low rates
- Locally optimal
  - Distortion decreases each step
- Training computationally expensive: not in real time
  - Iterative solutions only

- Many variants:
  - Multi-stage
  - Tree
  - *Constrained-entropy* version (around 1990)

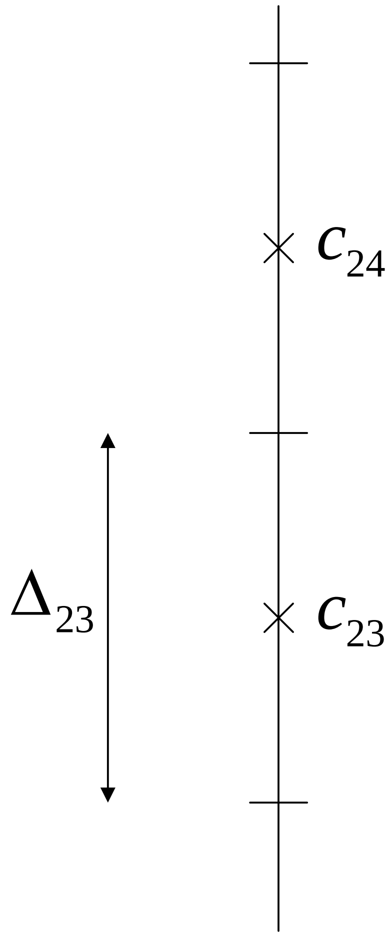
High-Rate Quantization

- Assume data density can be assumed constant within a cell (Bennett, 1948)
- Assume that notion *density of centroids* is meaningful
- Problem formulation
  - Given data density, distortion criterion, constraint
  - Find centroid density (“quantizer”)
- Advantage of approach
  - Optimal quantizer can be computed analytically
  - Can be done in real-time

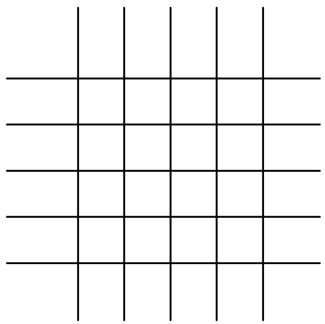
Distortion and Geometry: SQ

$$D_i = \frac{\int_{V_i} f_X(x)\, d(x, Q(x))\, dx}{\int_{v_i} f_X(x)\, dx} \approx \frac{f_X(x) \int_{V_i} (x - c_i)^2\, dx}{f_X(x)\Delta_i}$$

$$= \frac{1}{\Delta_i} \int_{-\Delta_i/2}^{\Delta_i/2} x^2\, dx = \frac{\Delta_i^2}{12}$$



- Scalar = cubic geometry



Distortion and Geometry: VQ

- Mean distortion in cell *i*, *r*'th power criterion, *per dim*

$$D_i = \frac{\int_{V_i} f_X(x^k) d(x^k, Q(x^k)) dx^k}{\int_{V_i} f_X(x^k) dx^k} \approx \frac{1}{kV_i} \int_{V_i} \left\| x^k - c_i^k \right\|_r dx^k$$

$$= V_i^{\frac{r}{k}} \frac{1}{k} V_i^{-\frac{r+k}{k}} \int_{V_i} \left\| x^k - c_i^k \right\|_r dx^k = V_i^{\frac{r}{k}} C(r,k,G(i))$$

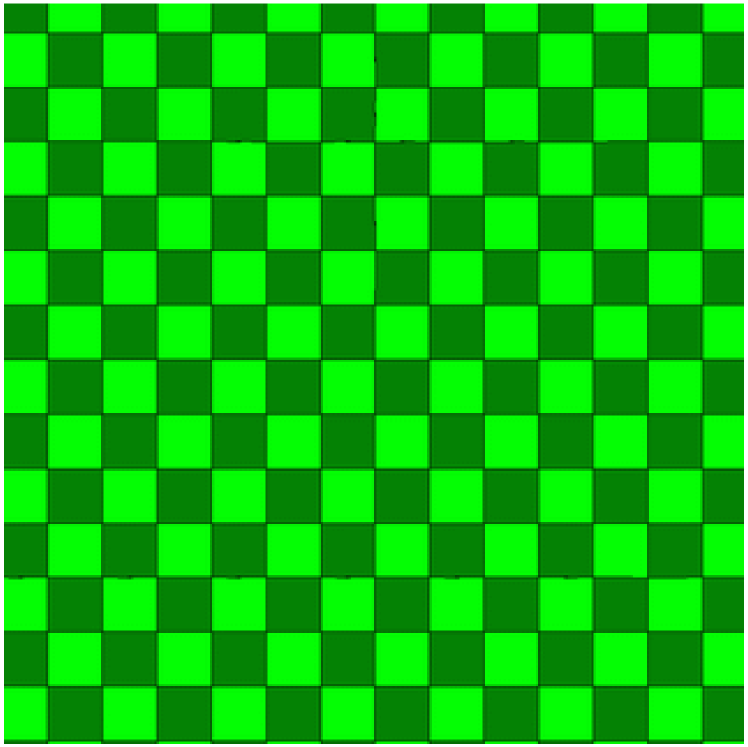
- $C(r,k,G(i)) \approx C(r,k,G(x^k))$ is the *inertial profile coefficient of quantization*

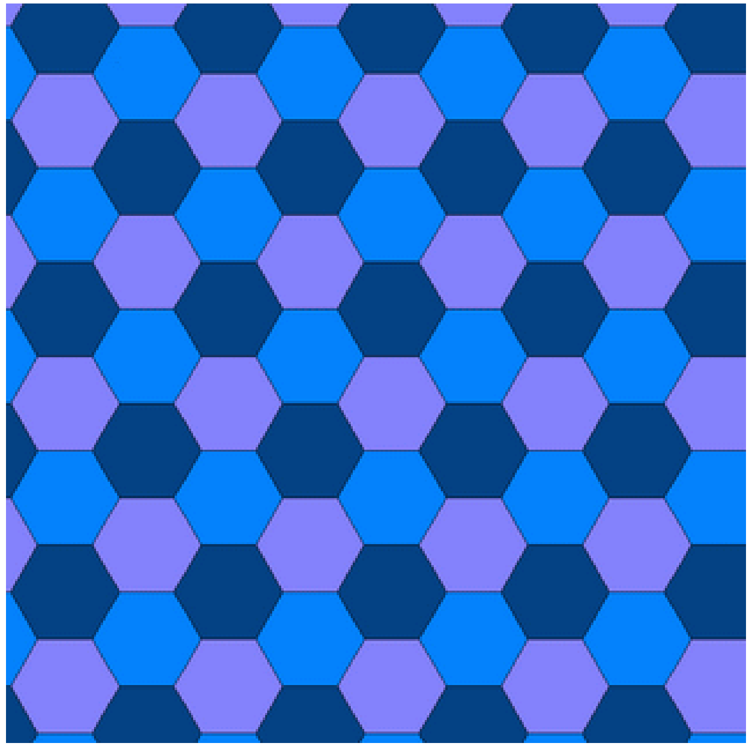
Quantization and Cell Geometry

- Scalar case: $$C(r=2, k=1, G=\text{optimal}) \approx \frac{1}{12} = 0.0833$$
  - cubic cells
- 2-D: $$C(r=2, k=2, G=\text{optimal}) \approx \frac{5}{36\sqrt{5}} = 0.0802$$
  - Hexagonal in sets of two dimensions
- ∞-D: $$C(r=2, k=\infty, G=\text{optimal}) \approx (2\pi e)^{-1} = 0.0585$$
  - Spherical cells
- VQ has space-filling advantage; 1.53 dB (= 0.25 bit)

CE Quantizers in 2D

- Two dimensions: square and hexagonal lattice





High-Rate Quantization

- What we have done:

  relate local geometry to local distortion

- Next step:

  to relate distortion, rate and centroid density

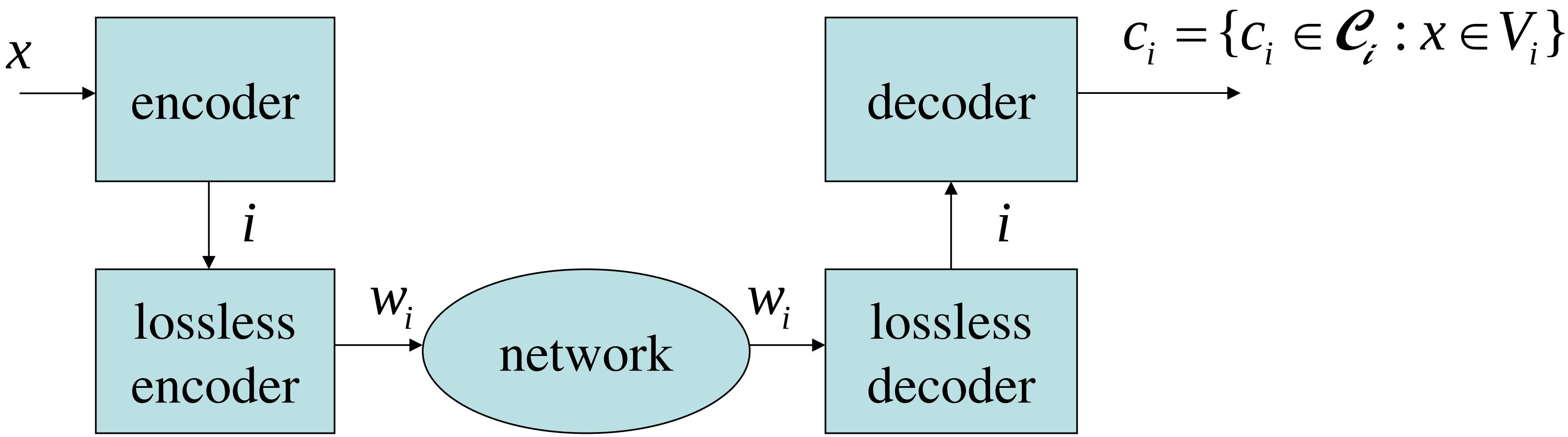
  (and local geometry)

- Centroid density: number of centroids/unit volume

  $g(x^k)$

Reminder: Constrained-Entropy Coding

- Apply lossless (entropy) coding to indices
  - Used to create ".zip"
- Rate is mean rate of codewords
  - Consistent with CDMA, packet networks, the future



Constrained-Entropy Quantization I

- Constraint on index entropy

$$
\begin{aligned}
H(I) &= -\sum_i p_I(i)\log(p_I(i)) \\
&= -\sum_i V_i f_X(c_i)\log(V_i f_X(c_i)) \\
&\approx -\int f_X(x^k)\Big(\log(f_X(x^k)) - \log(g_X(x^k)\Big)dx^k \\
&= h(X^k) + \int f_X(x^k)\log(g_X(x^k))dx^k
\end{aligned}
$$

- Equivalent constraint

$$
\int f_X(x^k)\log(g_X(x^k)dx^k = \text{constant}
$$

Constrained-Entropy Quantization II

- Distortion: $$D = \sum_{i} p_I(i) D_i = \sum_i p_I(i)\, V_i^{-\frac{r}{k}} C(r,k,G(i))$$

$$\approx C(r,k,G) \int f_X(x^k) g^{\frac{r}{k}}(x^k)\,dx$$

- Add Lagrange-multiplier term:

$$D \approx C(r,k,G) \int f_X(x^k) g^{-\frac{r}{k}}(x^k)\,dx$$

$$= C(r,k,G) \int f_X(x^k) \left(g^{-\frac{r}{k}}(x^k) + \lambda \log(g(x^k)) \right) dx$$

- Minimize; get Euler-Lagrange equation; solve

$$f_X(x^k) \left(g^{-\frac{r+k}{k}}(x^k) + \lambda g^{-1}(x^k) \right) = 0 \quad \rightarrow \quad \boxed{g(x^k) = \text{constant!!}}$$

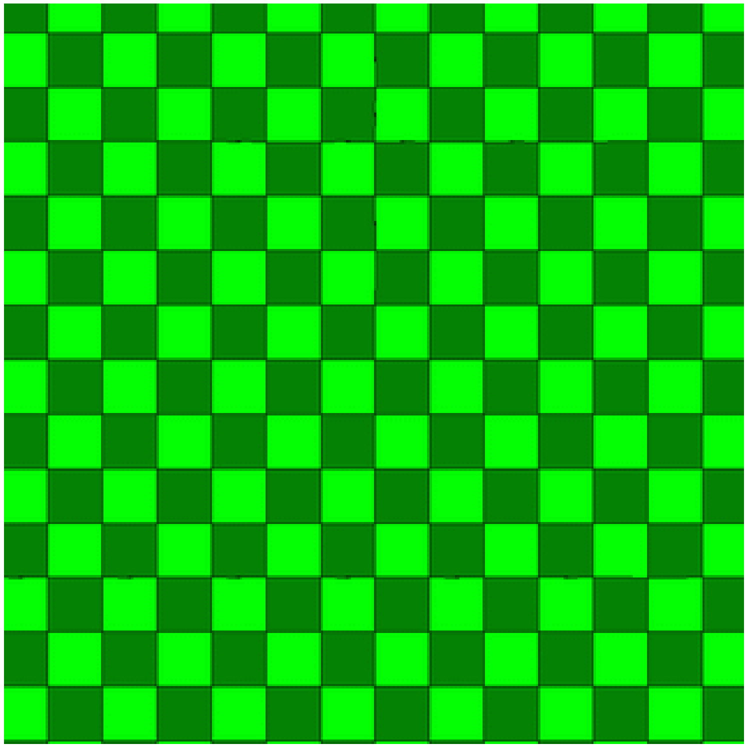
Moral of the Story

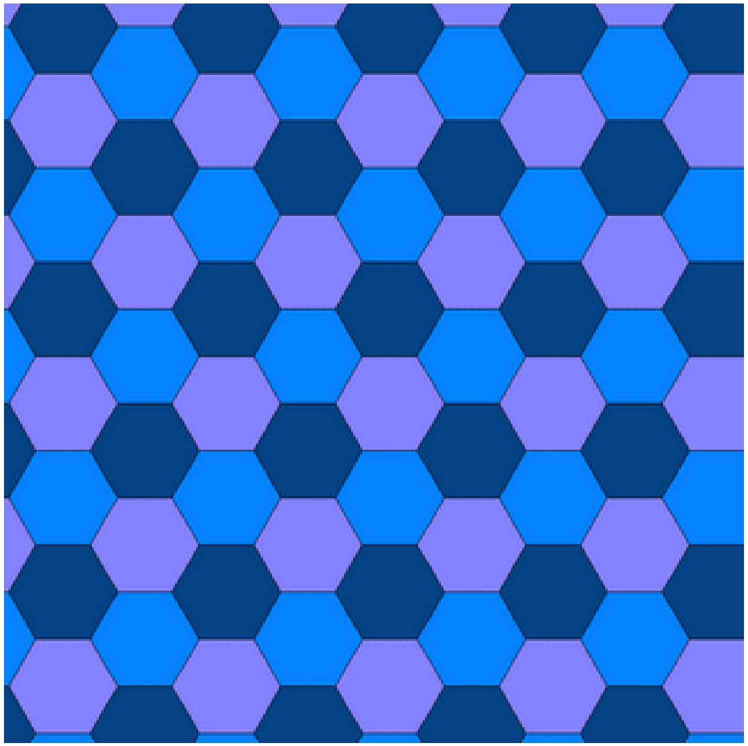
- For constrained-entropy quantization:
  simplest quantizer is best
  - All cells are same size and shape (not proven, that)
  - Facilitates low computational complexity quantizer
  - Can compute quantizer for given pdf and distortion
  - Does **not** mean entire encoder is low complexity!

- Somewhat non-intuitive:
  - Infinite number of cells/centroids!
  - Cell size independent of data density

CE Quantizers in 2D

- Two dimensions: square and hexagonal lattice





Constrained-Entropy Quantization IIa

- Complete solution:

$$\boxed{g(x^k) = \exp\left(H(I) - \mathrm{h}(\mathrm{X}^k)\right)}$$

- At a given distortion level the optimal centroid density:
  - increases with mean index rate
  - decreases with differential entropy of data (= complexity of data)
- Can adjust coder in real time!

Distortion-Rate Relation

- Relation distortion and rate (per dimension):

$$D = \sum_i p_I(i) D_i \approx C(r,k,G) \int f_X(x^k) g(x^k)^{-\frac{r}{k}} dx^k$$

$$= C(r,k,G) g(x^k)^{-\frac{r}{k}} \int f_X(x^k) dx^k$$

$$= C(r,k,G) \exp\left(-\frac{r}{k} \left(H(I) - h(X^k) \right) \right)$$

The Vector-Quantization Advantage

- Divide distortions of SQ and VQ (Gray & Lookabough, 1989)

$$\frac{D_{\mathrm{SQ}}}{D_{\mathrm{VQ}}} = \frac{C(r,1,G)\exp\left(-r\left(H(I)-h(X^1)\right)\right)}{C(r,k,G)\exp\left(-\frac{r}{k}\left(H(I)-h(X^k)\right)\right)}$$

$$= \frac{C(r,1,G)}{C(r,k,G)}\exp\left(r\left(h(X^1)-\frac{1}{k}h(X^k)\right)\right)$$

- Space-filling advantage

- Memory advantage (due to redundancy)

$$\rho = h(X^1) - \frac{1}{k}h(X^k)$$

Constrained-Entropy Quantization is Easy

- Uniform quantizer:
  - simple to implement
  - Small advantage from using best lattice
    - Somewhat more complicated

- Lossless coding is not easy:
  - Does not even exist in old-school quantization
  - Must know data density

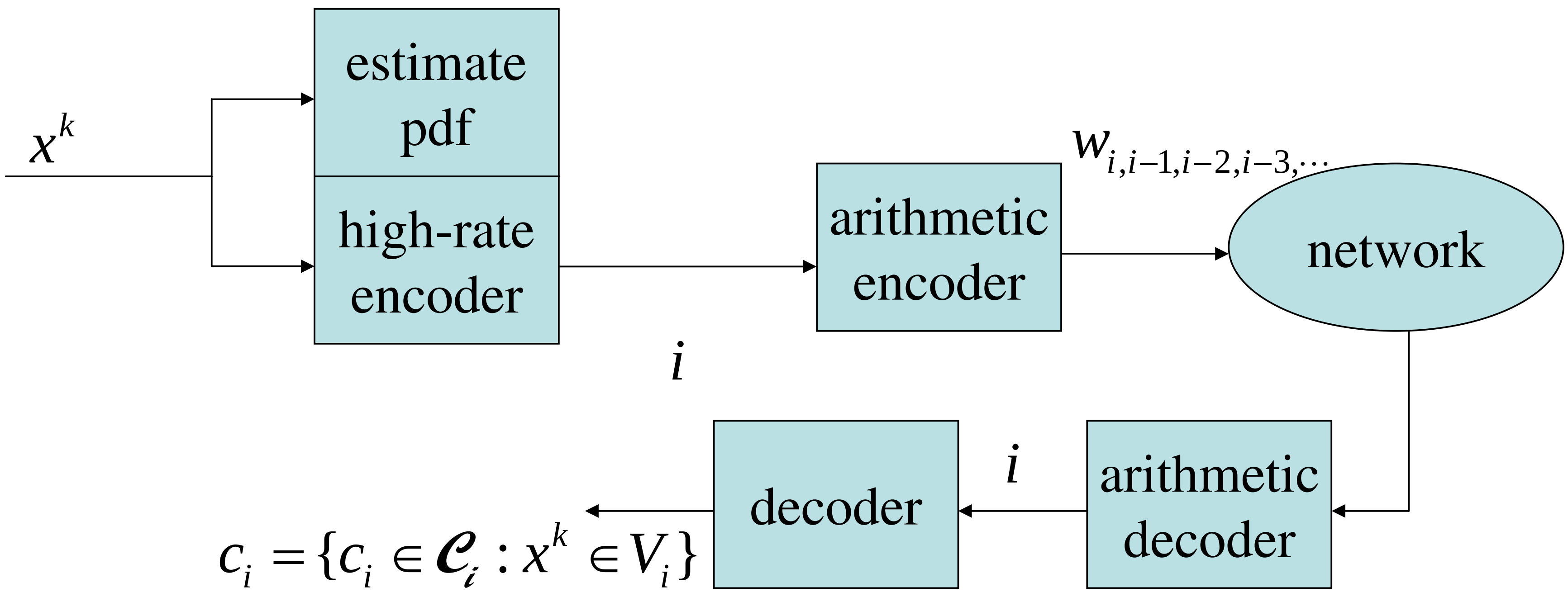
Some Notes on Lossless Coding

- Lossless coding tries to reduce rate to index entropy
- Huffman code:
  - Table $i \rightarrow w_i$ based on probability distribution
  - Works on per-variable basis; high overhead
  - Simple to implement

- Arithmetic code:
  - Computes codewords for sequence of coefficients
    - Tricky to write program
    - Low overhead
  - Requires cumulative distribution function (cmf)
    - Often nontrivial to obtain cmf
  - Preferred method

Practical High-Rate CE Coding

- No significant commercial implementations as yet
- Quantizer and arithmetic coder are *computed; flexible*

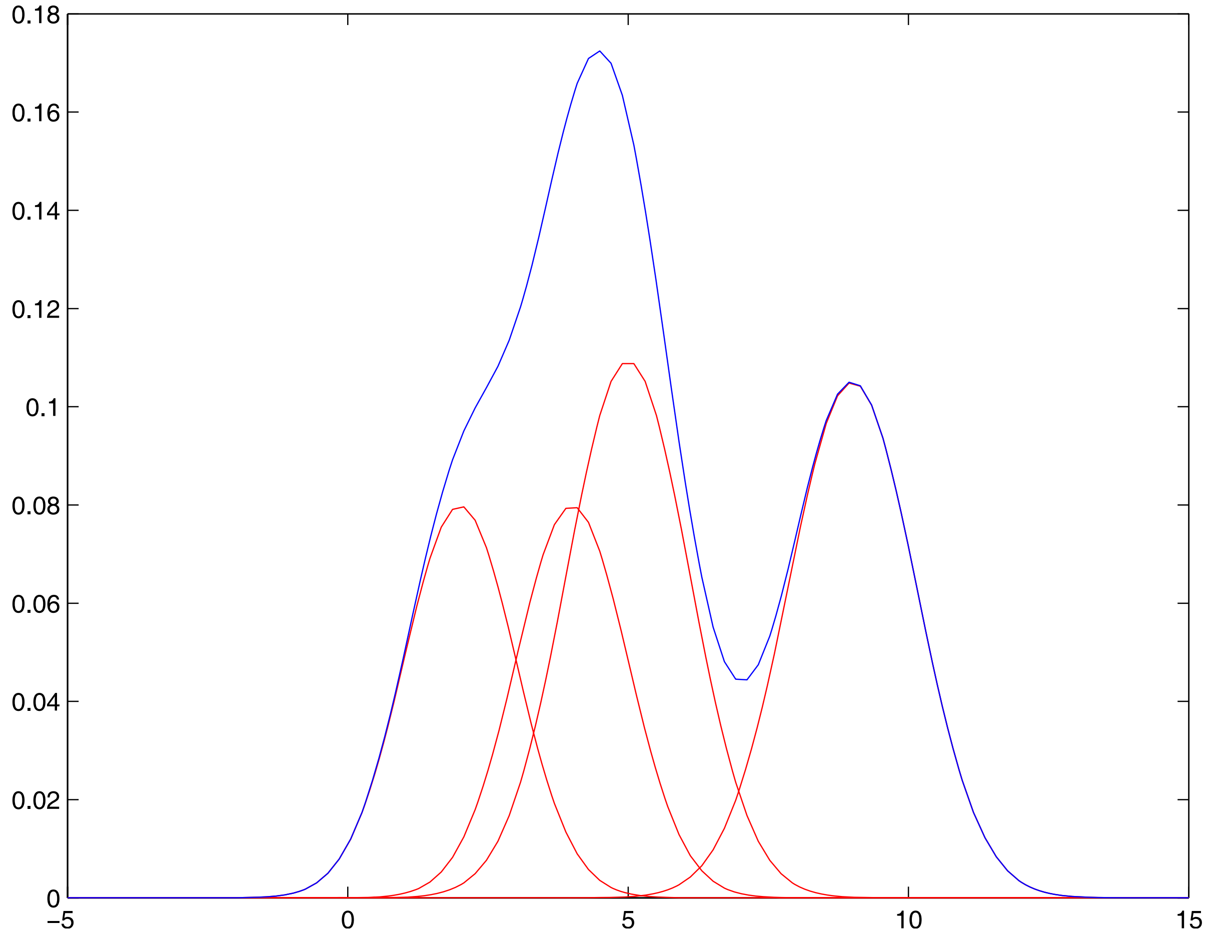


Example PDF Estimation

- Difficult; simplify problem:
  - Density modeled as mixture $p_X(x^k) = \sum_m p_M(m) p_{X,m}(x^k)$
    - Interpretation: data fall in one of set of probabilities
  - Each mixture component is Gaussian (usually)
  - Know how to design quantizer for Gaussian
    - Symmetric
    - Just one design procedure needed for cmf computation
  - Encode which component you select then use corresponding quantizer

Gaussian mixture

- Four components:



High-Rate Quantization

- Not yet widely applied
  - Real-time adaptation not used
- Constrained entropy (constraint on average rate)

What Have We Learned

- Problem:
  - Have audio or video data (transformed or not)
  - Need to encode efficiently
- Old-School Solution
  - Good performance / not flexible
  - Constrained resolution
  - Codebook (often computationally expensive)
  - Commonly used
- New-World Approach
  - Good performance / can adapt in real-time
  - Constrained entropy; requires lossless coder (arithmetic coder)
  - Quantizer and arithmetic coder computed = flexible
  - Not yet ready

Quantization Conclusions

- Emphasis was on performance
- Emphasis is on flexibility (but no loss of performance)