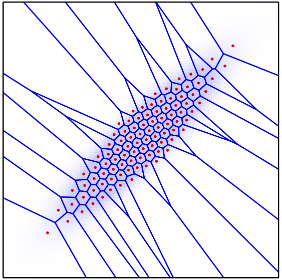
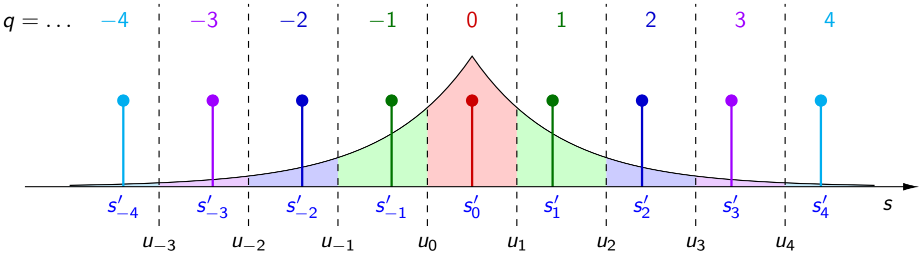
Vector Quantization



Last Lectures: Scalar Quantization



- **Performance of Scalar Quantizers**: Distortion (MSE) and Bit Rate

$$D = \mathrm{E}\Big\{ \big(S - Q(S) \big)^2 \Big\} = \sum_{\forall k} \int_{u_k}^{u_{k+1}} (s - s'_k)^2 \, f(s) \, \mathrm{d}s$$

$$R = \mathrm{E}\Big\{ \ell\big(Q(S) \big) \Big\} = \sum_{\forall k} \ell_k \int_{u_k}^{u_{k+1}} f(s) \, \mathrm{d}s$$

Last Lectures: Optimal Scalar Quantization

Lloyd Quantizer

- Minimizes distortion D for given number K of reconstruction levels
- Two optimization criterions:
  - Centroid condition (MSE): $s'_k = \mathrm{E}\{\, S \,|\, S \in \mathcal{I}_k \,\}$
  - Nearest neighbor condition (MSE): $u_k = (s'_k + s'_{k-1})/2$
- Lloyd quantizer design: Iterate between the two optimization criterions

Last Lectures: Optimal Scalar Quantization

Lloyd Quantizer

- Minimizes distortion D for given number K of reconstruction levels
- Two optimization criterions:
  - Centroid condition (MSE): $s'_k = \mathrm{E}\{\, S \,|\, S \in \mathcal{I}_k \,\}$
  - Nearest neighbor condition (MSE): $u_k = (s'_k + s'_{k-1})/2$
- Lloyd quantizer design: Iterate between the two optimization criterions

Entropy-Constrained Lloyd Quantizer

- Minimizes rate-distortion cost $J = D + \lambda R$ for given Lagrange multiplier $\lambda > 0$
- Three optimization criterions:
  - Centroid condition (MSE): $s'_k = \mathrm{E}\{\, S \,|\, S \in \mathcal{I}_k \,\}$
  - Entropy condition: $\ell_k = -\log_2 \int_{u_k}^{u_{k+1}} f(s)\, \mathrm{d}s$
  - Mod. nearest neighbor condition (MSE): $u_k = (s'_k + s'_{k-1})/2 + (\lambda/2)(\ell_k - \ell_{k-1})/(s'_k + s'_{k-1})$
- EC-Lloyd quantizer design: Iterate between the optimization criterions

Last Lectures: Performance of Scalar Quantizers

High-Rate Approximations (MSE Distortion)

- General form of high-rate distortion-rate function

$$D_X(R) = \varepsilon_X^2 \cdot \sigma^2 \cdot 2^{-2R}$$

where the constant factor ε_X^2 depends on shape of pdf and quantizer design

- ➡ Lloyd + fixed length: $\varepsilon_F^2 = \frac{1}{12} \left(\int_{-\infty}^{\infty} \sqrt[3]{f(s/\sigma)}\, \mathrm{d}s \right)^3$
- ➡ EC-Lloyd + variable length: $\varepsilon_V^2 = \frac{1}{12}\, 2^{2\,h(S/\sigma)}$ with $h(S) = -\int_{-\infty}^{\infty} f(s) \log_2 f(s)\, \mathrm{d}s$
- ➡ Shannon lower bound: $\varepsilon_L^2 = \frac{1}{2\pi e}\, 2^{2\,h(S/\sigma)}$

Last Lectures: Performance of Scalar Quantizers

High-Rate Approximations (MSE Distortion)

- General form of high-rate distortion-rate function

$$D_X(R) = \varepsilon_X^2 \cdot \sigma^2 \cdot 2^{-2R}$$

where the constant factor ε_X^2 depends on shape of pdf and quantizer design

➡ Lloyd + fixed length: $\varepsilon_F^2 = \frac{1}{12} \left(\int_{-\infty}^{\infty} \sqrt[3]{f(s/\sigma)} \, \mathrm{d}s \right)^3$

➡ EC-Lloyd + variable length: $\varepsilon_V^2 = \frac{1}{12} \, 2^{2\,h(S/\sigma)}$ with $h(S) = -\int_{-\infty}^{\infty} f(s) \log_2 f(s) \, \mathrm{d}s$

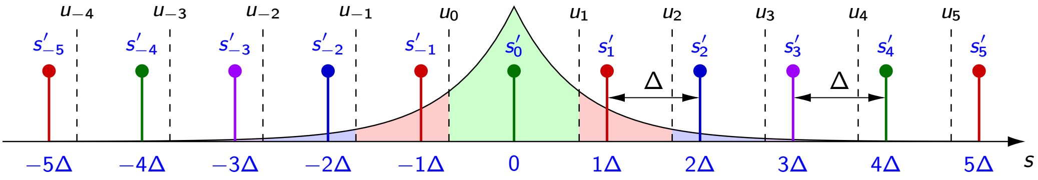
➡ Shannon lower bound: $\varepsilon_L^2 = \frac{1}{2\pi e} \, 2^{2\,h(S/\sigma)}$

Comparison of Coding Efficiency

- EC-Lloyd often significantly better than Lloyd (Gauss: 2.81 dB; Laplace: 5.63 dB)
- Constant high-rate performance gap between EC-Lloyd and Shannon lower bound

$$\frac{D_V(R)}{D_L(R)} = \frac{\pi e}{6} \approx 1.42 \;\; (1.53 \text{ dB}), \qquad R_V(D) - R_L(D) = \frac{1}{2} \log_2 \frac{\pi e}{6} \approx 0.25$$

Last Lectures: Scalar Quantization in Practice



Uniform Reconstruction Quantizers (URQs)

- Simple decoding process: $s'_n = \Delta \cdot q_n$
- Encoder can choose trade-off between coding efficiency and complexity
  - ➜ Simplest encoding: $q_n = \mathrm{round}\,(s_n/\Delta)$
  - ➜ Optimal encoding: Choose q_n that minimizes Lagrange cost $J(q_n) = (s_n - q_n \cdot \Delta)^2 + \lambda \cdot \ell_k$, typically using fixed relationship $\lambda = \mathrm{const} \cdot \Delta^2$
- URQs with optimal encoding are virtually as good as optimal scalar quantizers (for typical pdfs)

Quantization: Open Questions

Performance Gap to Theoretical Bound

- Remember: High-rate performance of optimal scalar quantizer for IID sources

$$\frac{D_V}{D_L}(R) = \frac{\pi e}{6} \approx 1.42 \qquad (1.53\,\text{dB loss in SNR})$$

➡ What causes this performance gap?

➡ How can the quantizer performance be improved?

Quantization: Open Questions

Performance Gap to Theoretical Bound

- Remember: High-rate performance of optimal scalar quantizer for IID sources

$$\frac{D_V}{D_L}(R) = \frac{\pi e}{6} \approx 1.42 \qquad (1.53\,\text{dB loss in SNR})$$

➡ What causes this performance gap?

➡ How can the quantizer performance be improved?

Quantization of Sources with Memory

- Scalar quantizers cannot exploit dependencies between samples (use only marginal pdf)

➡ How can we improve lossy coding for sources with memory?

Quantization: Open Questions

Performance Gap to Theoretical Bound

- Remember: High-rate performance of optimal scalar quantizer for IID sources

$$\frac{D_V}{D_L}(R) = \frac{\pi e}{6} \approx 1.42 \qquad \text{(1.53 dB loss in SNR)}$$

➡ What causes this performance gap?

➡ How can the quantizer performance be improved?

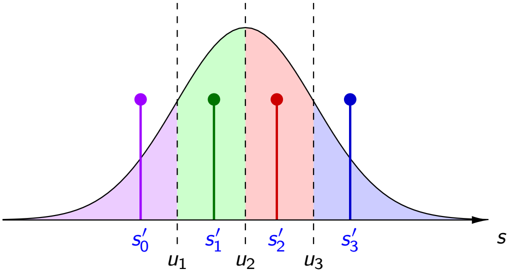
Quantization of Sources with Memory

- Scalar quantizers cannot exploit dependencies between samples (use only marginal pdf)

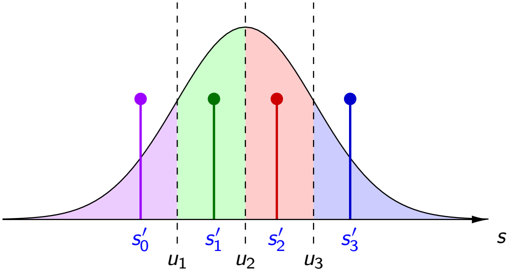
➡ How can we improve lossy coding for sources with memory?

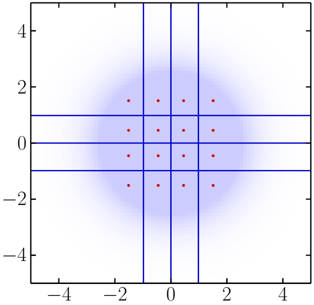
  - Conditional entropy coding of quantization indexes ?
  - Combination of scalar quantization and prediction ?
  - ... ?

Scalar Quantizers in N-dimensional Signal Space



Scalar Quantizers in N-dimensional Signal Space

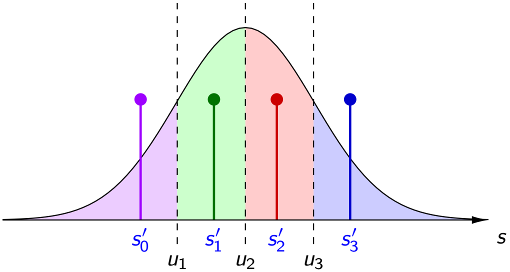


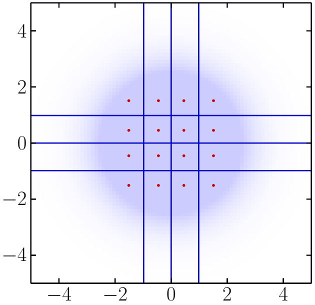


Interpretation of Scalar Quantization in N-dimensional Signal Space

- N-dimensional input vector $\boldsymbol{s}$ is mapped to N-dimensional reconstructed vector $\boldsymbol{s}'$
- All vectors $\boldsymbol{s}$ inside a quantization cell $\mathcal{C}_k$ are mapped to the same reconstruction vector $\boldsymbol{s}'_k$

Scalar Quantizers in N-dimensional Signal Space



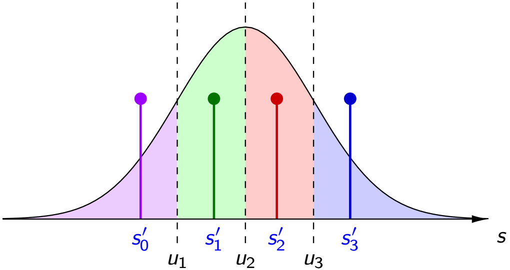


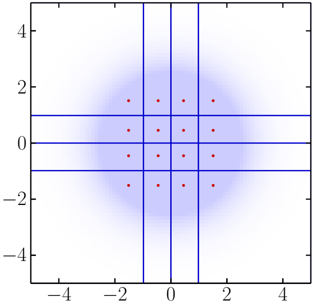
Interpretation of Scalar Quantization in N-dimensional Signal Space

- N-dimensional input vector $\boldsymbol{s}$ is mapped to N-dimensional reconstructed vector $\boldsymbol{s'}$
- All vectors $\boldsymbol{s}$ inside a quantization cell $\mathcal{C}_k$ are mapped to the same reconstruction vector $\boldsymbol{s}'_k$

➡ Quantization cells $\mathcal{C}_k$ form hyper-rectangles in N-dimensional signal space

Scalar Quantizers in N-dimensional Signal Space





Interpretation of Scalar Quantization in N-dimensional Signal Space

- N-dimensional input vector $\boldsymbol{s}$ is mapped to N-dimensional reconstructed vector $\boldsymbol{s}'$
- All vectors $\boldsymbol{s}$ inside a quantization cell $\mathcal{C}_k$ are mapped to the same reconstruction vector $\boldsymbol{s}'_k$

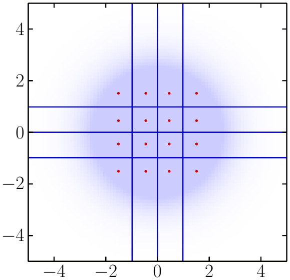
➡ Quantization cells $\mathcal{C}_k$ form hyper-rectangles in N-dimensional signal space

➡ Reconstruction vectors $\boldsymbol{s}'_k$ lie on orthogonal grid aligned with coordinate axes

Vector Quantization: Relaxing Structural Constraints

scalar quantizer

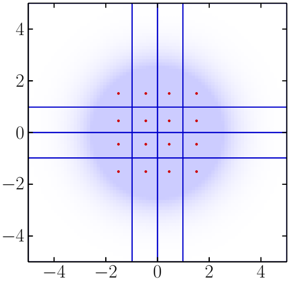
(dimension $N = 1$)



Vector Quantization: Relaxing Structural Constraints

scalar quantizer

(dimension $N = 1$)



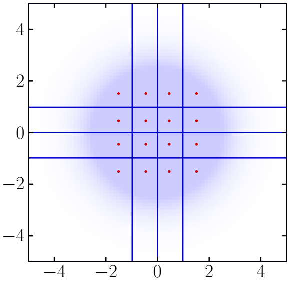
Vector Quantization

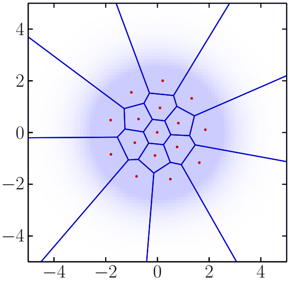
- Joint quantization of vectors/blocks $\boldsymbol{s}$ of $N > 1$ successive input samples

Vector Quantization: Relaxing Structural Constraints

scalar quantizer

(dimension $N = 1$)





vector quantizer

(dimension $N = 2$)

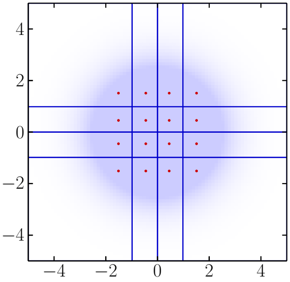
Vector Quantization

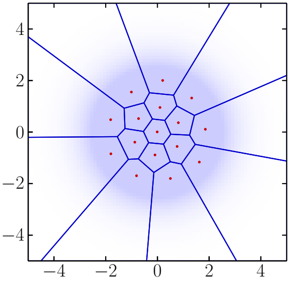
- Joint quantization of vectors/blocks $\boldsymbol{s}$ of $N > 1$ successive input samples

Vector Quantization: Relaxing Structural Constraints

scalar quantizer

(dimension $N = 1$)





vector quantizer

(dimension $N = 2$)

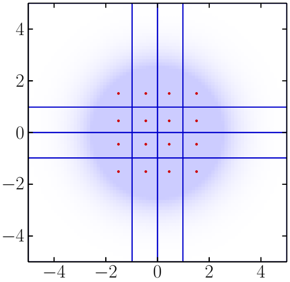
Vector Quantization

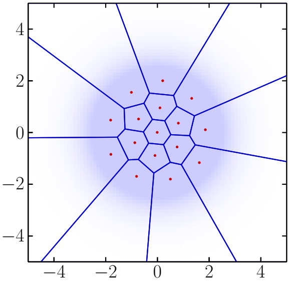
- Joint quantization of vectors/blocks $\boldsymbol{s}$ of $N > 1$ successive input samples
- Relax structural constraints that are implicitly imposed by scalar quantization
  - Quantization cells $\mathcal{C}_k$ can be arbitrarily shaped in N-dimensional space
  - Reconstruction vectors $\boldsymbol{s}'_k$ can be arbitrarily placed in N-dimensional space

Vector Quantization: Relaxing Structural Constraints

scalar quantizer

(dimension $N = 1$)





vector quantizer

(dimension $N = 2$)

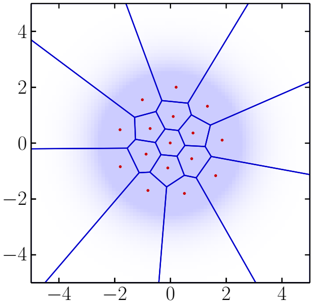
Vector Quantization

- Joint quantization of vectors/blocks $\boldsymbol{s}$ of $N > 1$ successive input samples
- Relax structural constraints that are implicitly imposed by scalar quantization
  - Quantization cells $\mathcal{C}_k$ can be arbitrarily shaped in N-dimensional space
  - Reconstruction vectors $\boldsymbol{s}'_k$ can be arbitrarily placed in N-dimensional space

➡ **Allows a number of new options in designing quantizers**

Structure of Vector Quantizers

Vector Quantizers of Quantizer Dimension N



- Map N-d input vectors $\boldsymbol{s}$ to N-d output vectors $\boldsymbol{s}'_k$

$$Q: \ \mathbb{R}^N \mapsto \{\, \boldsymbol{s}'_0, \, \boldsymbol{s}'_1, \, \boldsymbol{s}'_2, \, \cdots \}$$

- Partition N-d space into countable number of quantization cells $\mathcal{C}_k$

$$\mathcal{C}_k = \{\, \boldsymbol{s} \in \mathbb{R}^N : \ Q(\boldsymbol{s}) = \boldsymbol{s}'_k \,\}$$

- All input vectors $\boldsymbol{s}$ that fall inside a quantization cell $\mathcal{C}_k$ are mapped to the associated reconstruction vector $\boldsymbol{s}'_k$

Structure of Vector Quantizers

Vector Quantizers of Quantizer Dimension N

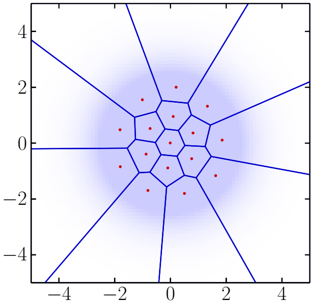
- Map N-d input vectors $\boldsymbol{s}$ to N-d output vectors $\boldsymbol{s}'_k$

$$Q : \mathbb{R}^N \mapsto \{ \boldsymbol{s}'_0, \boldsymbol{s}'_1, \boldsymbol{s}'_2, \cdots \}$$

- Partition N-d space into countable number of quantization cells $\mathcal{C}_k$

$$\mathcal{C}_k = \{ \boldsymbol{s} \in \mathbb{R}^N : \; Q(\boldsymbol{s}) = \boldsymbol{s}'_k \}$$

- All input vectors $\boldsymbol{s}$ that fall inside a quantization cell $\mathcal{C}_k$ are mapped to the associated reconstruction vector $\boldsymbol{s}'_k$



Vector Quantization and Entropy Coding

- Quantization index k indicates quantization cell $\mathcal{C}_k$ and reconstruction vector $\boldsymbol{s}'_k$
  - ➡ Encoder mapping: $\alpha(\boldsymbol{s}) = k, \quad \forall \boldsymbol{s} \in \mathcal{C}_k$
  - ➡ Decoder mapping: $\beta(k) = \boldsymbol{s}'_k$

Vector Quantization: Encoding and Decoding

- Arbitrarily shaped quantization cells $\mathcal{C}_k$ are difficult to store and check

➡ Concept of quantization cells is not required in practice

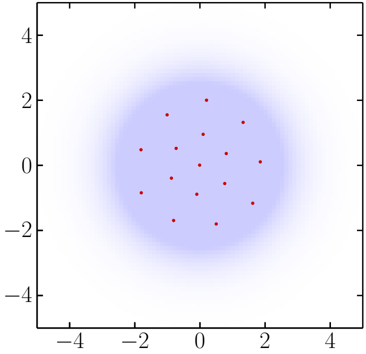
Vector Quantization: Encoding and Decoding

- Arbitrarily shaped quantization cells $\mathcal{C}_k$ are difficult to store and check

➡ Concept of quantization cells is not required in practice

Encoding

- Select the reconstruction vector $\boldsymbol{s}'_k$ that minimizes a **distance measure** d to the input vector $\boldsymbol{s}$



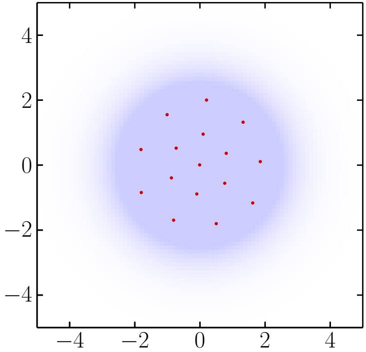
Vector Quantization: Encoding and Decoding

- Arbitrarily shaped quantization cells $\mathcal{C}_k$ are difficult to store and check

➡ Concept of quantization cells is not required in practice

Encoding

- Select the reconstruction vector $\boldsymbol{s}'_k$ that minimizes a **distance measure** d to the input vector $\boldsymbol{s}$
- Possible distance measures:
  - ➡ MSE distortion: $d = ||\boldsymbol{s} - \boldsymbol{s}'_k||_2^2$
  - ➡ Lagrangan cost: $d = ||\boldsymbol{s} - \boldsymbol{s}'_k||_2^2 + \lambda\,\ell_k$



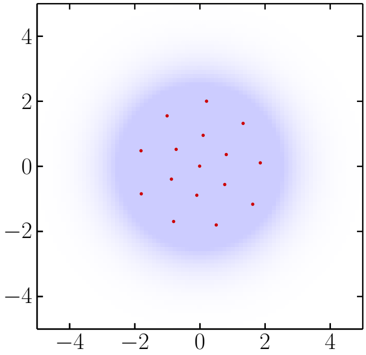
Vector Quantization: Encoding and Decoding

- Arbitrarily shaped quantization cells $\mathcal{C}_k$ are difficult to store and check

➡ Concept of quantization cells is not required in practice

Encoding

- Select the reconstruction vector $\boldsymbol{s}'_k$ that minimizes a **distance measure** d to the input vector $\boldsymbol{s}$
- Possible distance measures:
  - ➡ MSE distortion: $d = ||\boldsymbol{s} - \boldsymbol{s}'_k||_2^2$
  - ➡ Lagrangan cost: $d = ||\boldsymbol{s} - \boldsymbol{s}'_k||_2^2 + \lambda\, \ell_k$



Decoding

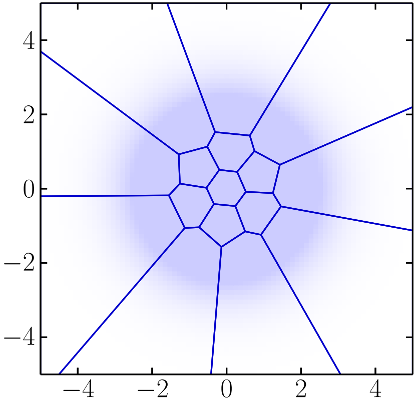
- Output reconstruction vector $\boldsymbol{s}'_k$ indicated by transmitted quantization index k (use array in decoder)

Performance of Vector Quantizers: Bit Rate

- Let ℓ_k be the codeword length for quantization index k

➡ **Average bit rate R per sample**

$$\boxed{R = \frac{1}{N}\,\mathrm{E}\Big\{\, \ell(\,Q(\boldsymbol{S})\,)\,\Big\} = \frac{1}{N}\sum_{\forall k} p_k\,\ell_k}$$



Performance of Vector Quantizers: Bit Rate

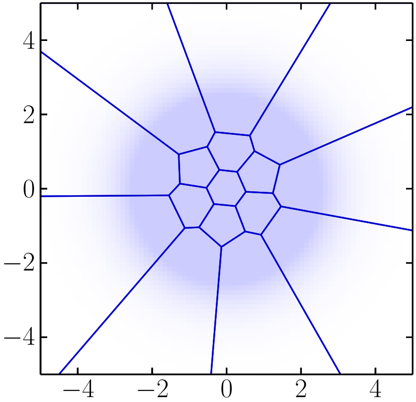
- Let ℓ_k be the codeword length for quantization index k

➡ **Average bit rate R per sample**

$$\boxed{R = \frac{1}{N} \mathrm{E}\Big\{ \ell(\,Q(\boldsymbol{S})\,) \Big\} = \frac{1}{N} \sum_{\forall k} p_k \, \ell_k}$$

- Probability p_k of quantization cell $\mathcal{C}_k$ / quant. index k

$$p_k = \int_{\mathcal{C}_k} f(\boldsymbol{s}) \, \mathrm{d}\boldsymbol{s}$$



Performance of Vector Quantizers: Bit Rate

- Let ℓ_k be the codeword length for quantization index k

➡ **Average bit rate R per sample**

$$\boxed{R = \frac{1}{N}\,\mathrm{E}\Big\{\, \ell(\,Q(\boldsymbol{S})\,)\,\Big\} = \frac{1}{N}\sum_{\forall k} p_k\,\ell_k}$$

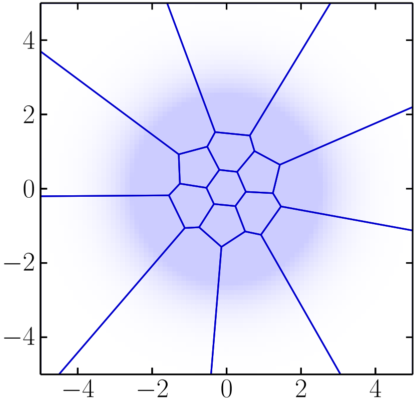
- Probability p_k of quantization cell $\mathcal{C}_k$ / quant. index k

$$p_k = \int_{\mathcal{C}_k} f(\boldsymbol{s})\,\mathrm{d}\boldsymbol{s}$$

- Approximation for training set

$$p_k = \frac{n(k)}{\sum_k n(k)}$$

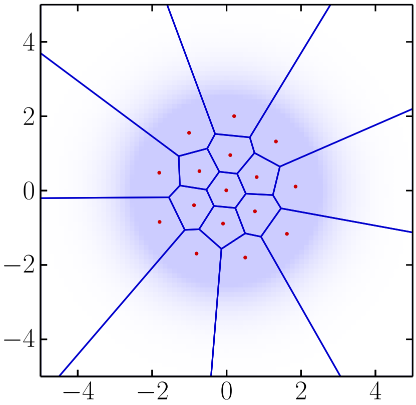
where $n(k)$ is the number of vectors assigned to $\mathcal{C}_k$ / $\boldsymbol{s}'_k$



Performance of Vector Quantizers: Distortion

➡ **Average MSE distortion D per sample**

$$D = \frac{1}{N}\,\mathrm{E}\Big\{ \, \big|\big|\, \boldsymbol{S} - Q(\boldsymbol{S}) \,\big|\big|_2^2 \, \Big\}$$

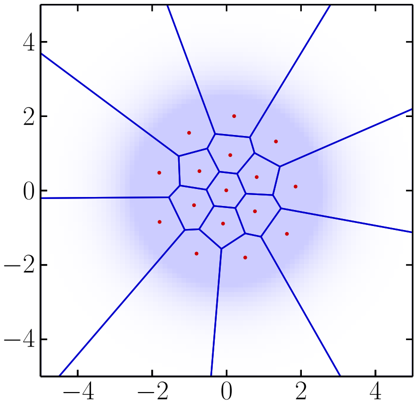


Performance of Vector Quantizers: Distortion

➡ **Average MSE distortion D per sample**

$$D = \frac{1}{N} \mathrm{E}\left\{ \, || \boldsymbol{S} - Q(\boldsymbol{S}) \, ||_2^2 \, \right\}$$

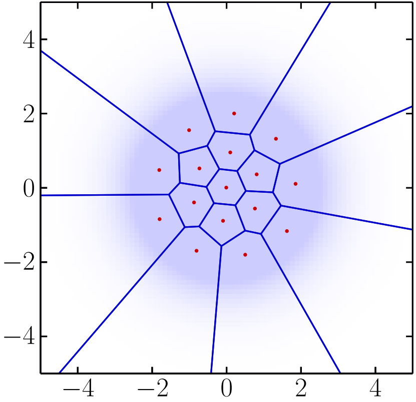
$$= \frac{1}{N} \int_{\mathbb{R}^N} || \boldsymbol{s} - Q(\boldsymbol{s}) \, ||_2^2 \, f(\boldsymbol{s}) \, \mathrm{d}\boldsymbol{s}$$



Performance of Vector Quantizers: Distortion

➡ Average MSE distortion D per sample

$$
\begin{aligned}
D &= \frac{1}{N}\,\mathrm{E}\Big\{\, \big|\big|\, \boldsymbol{S} - Q(\boldsymbol{S}) \,\big|\big|_2^2 \,\Big\} \\
&= \frac{1}{N} \int_{\mathbb{R}^N} \big|\big|\, \boldsymbol{s} - Q(\boldsymbol{s}) \,\big|\big|_2^2 \, f(\boldsymbol{s})\, \mathrm{d}\boldsymbol{s} \\
&= \frac{1}{N} \sum_{\forall k} \int_{\mathcal{C}_k} \big|\big|\, \boldsymbol{s} - \boldsymbol{s}'_{\boldsymbol{k}} \,\big|\big|_2^2 \, f(\boldsymbol{s})\, \mathrm{d}\boldsymbol{s}
\end{aligned}
$$



Performance of Vector Quantizers: Distortion

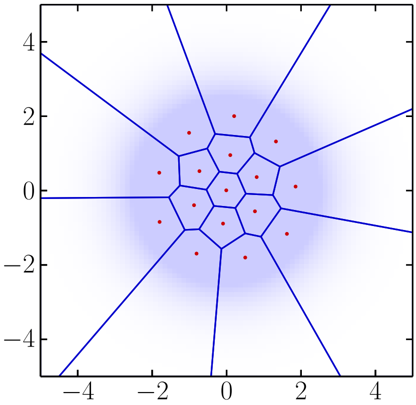
➡ **Average MSE distortion D per sample**

$$D = \frac{1}{N} \mathrm{E}\Big\{ \, || \, \boldsymbol{S} - Q(\boldsymbol{S}) \, ||_2^2 \, \Big\}$$

$$= \frac{1}{N} \int_{\mathbb{R}^N} || \, \boldsymbol{s} - Q(\boldsymbol{s}) \, ||_2^2 \, f(\boldsymbol{s}) \, \mathrm{d}\boldsymbol{s}$$

$$= \frac{1}{N} \sum_{\forall k} \int_{\mathcal{C}_k} || \, \boldsymbol{s} - \boldsymbol{s}'_k \, ||_2^2 \, f(\boldsymbol{s}) \, \mathrm{d}\boldsymbol{s}$$

$$\boxed{D = \frac{1}{N} \sum_{\forall k} \int_{\mathcal{C}_k} \left(\boldsymbol{s} - \boldsymbol{s}'_k \right)^{\mathrm{T}} \left(\boldsymbol{s} - \boldsymbol{s}'_k \right) \, f(\boldsymbol{s}) \, \mathrm{d}\boldsymbol{s}}$$



Performance of Vector Quantizers: Distortion

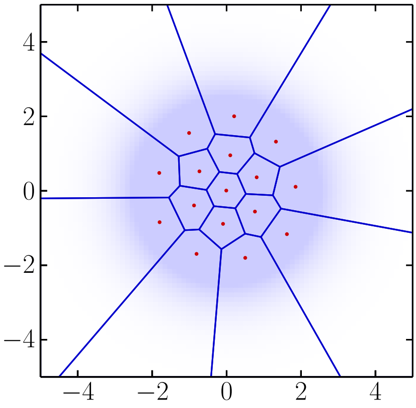
➡ **Average MSE distortion D per sample**

$$D = \frac{1}{N} \mathrm{E}\Big\{ \,||\, \boldsymbol{S} - Q(\boldsymbol{S}) \,||_2^2 \,\Big\}$$

$$= \frac{1}{N} \int_{\mathbb{R}^N} ||\, \boldsymbol{s} - Q(\boldsymbol{s}) \,||_2^2 \, f(\boldsymbol{s}) \, \mathrm{d}\boldsymbol{s}$$

$$= \frac{1}{N} \sum_{\forall k} \int_{\mathcal{C}_k} ||\, \boldsymbol{s} - \boldsymbol{s}'_k \,||_2^2 \, f(\boldsymbol{s}) \, \mathrm{d}\boldsymbol{s}$$

$$\boxed{D = \frac{1}{N} \sum_{\forall k} \int_{\mathcal{C}_k} \left(\boldsymbol{s} - \boldsymbol{s}'_k\right)^{\mathrm{T}} \left(\boldsymbol{s} - \boldsymbol{s}'_k\right) \, f(\boldsymbol{s}) \, \mathrm{d}\boldsymbol{s}}$$



- Approximation for training set $\{\boldsymbol{s}_n\}$ of L vectors

$$D = \frac{1}{L} \sum_{\forall n} ||\boldsymbol{s}_n - Q(\boldsymbol{s}_n)||_2^2$$

Optimal Vector Quantizer for Fixed-Length Coding

Goal: Minimize MSE Distortion for K Quantization Cells

- Similar to Scalar Lloyd Quantizer
- Neglect impact of entropy coding ➡ Consider fixed-length coding

Optimal Vector Quantizer for Fixed-Length Coding

Goal: Minimize MSE Distortion for K Quantization Cells

- Similar to Scalar Lloyd Quantizer
- Neglect impact of entropy coding ➡ Consider fixed-length coding

➡ Rate R and MSE distortion D are given by

$$R = \frac{1}{N} \left\lceil \log_2 K \right\rceil \qquad \text{(typically } K = 2^B\text{, with } B \text{ being the bits per codeword)}$$

$$D = \frac{1}{N} \sum_{\forall k} \int_{\mathcal{C}_k} \left|\left| \boldsymbol{s} - \boldsymbol{s}'_k \right|\right|_2^2 f(\boldsymbol{s}) \, \mathrm{d}\boldsymbol{s} = \frac{1}{N} \sum_{\forall k} \int_{\mathcal{C}_k} \left(\boldsymbol{s} - \boldsymbol{s}'_k \right)^{\mathrm{T}} \left(\boldsymbol{s} - \boldsymbol{s}'_k \right) f(\boldsymbol{s}) \, \mathrm{d}\boldsymbol{s}$$

Optimal Vector Quantizer for Fixed-Length Coding

Goal: Minimize MSE Distortion for K Quantization Cells

- Similar to Scalar Lloyd Quantizer
- Neglect impact of entropy coding ➡ Consider fixed-length coding

➡ Rate R and MSE distortion D are given by

$$R = \frac{1}{N} \left\lceil \log_2 K \right\rceil \qquad \text{(typically } K = 2^B\text{, with } B \text{ being the bits per codeword)}$$

$$D = \frac{1}{N} \sum_{\forall k} \int_{\mathcal{C}_k} \left|\left| \boldsymbol{s} - \boldsymbol{s}'_{\boldsymbol{k}} \right|\right|_2^2 f(\boldsymbol{s}) \, \mathrm{d}\boldsymbol{s} = \frac{1}{N} \sum_{\forall k} \int_{\mathcal{C}_k} \left(\boldsymbol{s} - \boldsymbol{s}'_{\boldsymbol{k}} \right)^{\mathrm{T}} \left(\boldsymbol{s} - \boldsymbol{s}'_{\boldsymbol{k}} \right) f(\boldsymbol{s}) \, \mathrm{d}\boldsymbol{s}$$

Optimize Quantizer of size K

- Derive necessary conditions for optimality (similar to Lloyd quantizer)
- Construct iterative algorithm for designing quantizer

Optimality Conditions for Fixed-Length Coding

Necessary Conditions for Optimality (MSE distortion)

1. Centroid condition (for reconstruction vectors $\boldsymbol{s}'_k$)

$$\boxed{\boldsymbol{s}'_k = \mathrm{E}\{\, \boldsymbol{S} \,|\, \boldsymbol{S} \in \mathcal{C}_k \,\} = \frac{1}{p_k} \int_{\mathcal{C}_k} \boldsymbol{s}\, f(\boldsymbol{s})\, \mathrm{d}\boldsymbol{s}}$$

Optimality Conditions for Fixed-Length Coding

Necessary Conditions for Optimality (MSE distortion)

1. Centroid condition (for reconstruction vectors $\boldsymbol{s}'_k$)

$$\boxed{\boldsymbol{s}'_k = \mathrm{E}\{\ \boldsymbol{S} \,|\, \boldsymbol{S} \in \mathcal{C}_k\ \} = \frac{1}{p_k} \int_{\mathcal{C}_k} \boldsymbol{s}\, f(\boldsymbol{s})\, \mathrm{d}\boldsymbol{s}}$$

➡ Centroid condition for training set

$$\boldsymbol{s}'_k = \frac{1}{n(k)} \sum_{\forall \boldsymbol{s}:\, \alpha(\boldsymbol{s})=k} \boldsymbol{s} \qquad \text{with} \qquad n(k) = \sum_{\forall \boldsymbol{s}:\, \alpha(\boldsymbol{s})=k} 1$$

Optimality Conditions for Fixed-Length Coding

Necessary Conditions for Optimality (MSE distortion)

1 Centroid condition (for reconstruction vectors $\boldsymbol{s}'_k$)

$$\boxed{\boldsymbol{s}'_k = \mathrm{E}\{\, \boldsymbol{S} \,|\, \boldsymbol{S} \in \mathcal{C}_k \,\} = \frac{1}{p_k} \int_{\mathcal{C}_k} \boldsymbol{s}\, f(\boldsymbol{s})\, \mathrm{d}\boldsymbol{s}}$$

➡ Centroid condition for training set

$$\boldsymbol{s}'_k = \frac{1}{n(k)} \sum_{\forall \boldsymbol{s}:\, \alpha(\boldsymbol{s}) = k} \boldsymbol{s} \qquad \text{with} \qquad n(k) = \sum_{\forall \boldsymbol{s}:\, \alpha(\boldsymbol{s}) = k} 1$$

2 Nearest neighbour condition (for quantization cells $\mathcal{C}_k$ / encoder mapping $\alpha(.)$)

$$\boxed{\alpha(\boldsymbol{s}) = \arg \min_{\forall k} \left|\left| \boldsymbol{s} - \boldsymbol{s}'_k \right|\right|_2^2}$$

The Linde-Buzo-Gray (LBG) Algorithm for a Training Set (MSE Distortion)

Given is:

- the dimension N and the size K of the quantizer
- a sufficiently large realization $\{\boldsymbol{s}_n\}$ of considered source

The Linde-Buzo-Gray (LBG) Algorithm for a Training Set (MSE Distortion)

Given is:
- the dimension N and the size K of the quantizer
- a sufficiently large realization $\{\boldsymbol{s}_n\}$ of considered source

Iterative quantizer design

1. Choose an initial set of K reconstruction vectors $\{\boldsymbol{s}'_k\}$

The Linde-Buzo-Gray (LBG) Algorithm for a Training Set (MSE Distortion)

Given is:
- the dimension N and the size K of the quantizer
- a sufficiently large realization $\{\boldsymbol{s}_n\}$ of considered source

Iterative quantizer design

1. Choose an initial set of K reconstruction vectors $\{\boldsymbol{s}'_{\boldsymbol{k}}\}$
2. Associate all vectors of the training set $\{\boldsymbol{s}_n\}$ with one of the quantization cells $\mathcal{C}_k$

$$q(\boldsymbol{s}_n) = \arg\min_{\forall k} \left|\left|\boldsymbol{s}_n - \boldsymbol{s}'_{\boldsymbol{k}}\right|\right|_2^2 \qquad \text{(nearest neighbor condition)}$$

The Linde-Buzo-Gray (LBG) Algorithm for a Training Set (MSE Distortion)

Given is:
- the dimension N and the size K of the quantizer
- a sufficiently large realization $\{\boldsymbol{s}_n\}$ of considered source

Iterative quantizer design

1. Choose an initial set of K reconstruction vectors $\{\boldsymbol{s}'_k\}$
2. Associate all vectors of the training set $\{\boldsymbol{s}_n\}$ with one of the quantization cells $\mathcal{C}_k$

$$q(\boldsymbol{s}_n) = \arg\min_{\forall k} \left|\left|\boldsymbol{s}_n - \boldsymbol{s}'_k\right|\right|_2^2 \qquad \text{(nearest neighbor condition)}$$

3. Update the reconstruction vectors $\{\boldsymbol{s}'_k\}$ according to

$$\boldsymbol{s}'_k = \frac{1}{n(k)} \sum_{\forall n:\, q(\boldsymbol{s}_n)=k} \boldsymbol{s}_n \qquad \text{(centroid condition)}$$

where $n(k)$ is the number of sample vectors $\boldsymbol{s}_n$ assigned to $\mathcal{C}_k$

The Linde-Buzo-Gray (LBG) Algorithm for a Training Set (MSE Distortion)

Given is:
- the dimension N and the size K of the quantizer
- a sufficiently large realization $\{\boldsymbol{s}_n\}$ of considered source

Iterative quantizer design

1. Choose an initial set of K reconstruction vectors $\{\boldsymbol{s}'_{\boldsymbol{k}}\}$
2. Associate all vectors of the training set $\{\boldsymbol{s}_n\}$ with one of the quantization cells $\mathcal{C}_k$

$$q(\boldsymbol{s}_n) = \arg\min_{\forall k} \left|\left|\boldsymbol{s}_n - \boldsymbol{s}'_{\boldsymbol{k}}\right|\right|_2^2 \qquad \text{(nearest neighbor condition)}$$

3. Update the reconstruction vectors $\{\boldsymbol{s}'_{\boldsymbol{k}}\}$ according to

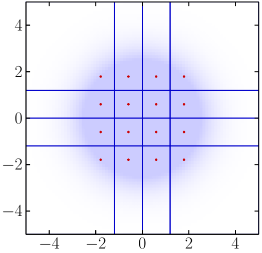
$$\boldsymbol{s}'_{\boldsymbol{k}} = \frac{1}{n(k)} \sum_{\forall n:\, q(\boldsymbol{s}_n)=k} \boldsymbol{s}_n \qquad \text{(centroid condition)}$$

   where $n(k)$ is the number of sample vectors $\boldsymbol{s}_n$ assigned to $\mathcal{C}_k$

4. Repeat the previous two steps until convergence

Example: LBG Algorithm for Gaussian IID $(\sigma^2 = 1,\ N = 2,\ K = 16)$

initialization

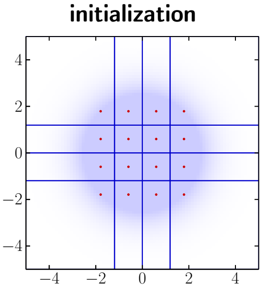


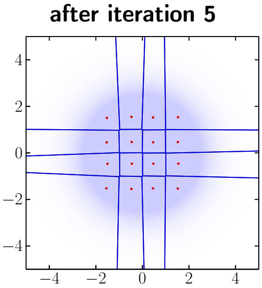
$$R = 2$$

$$D = 0.122$$

$$\text{SNR} = 9.12\,\text{dB}$$

Example: LBG Algorithm for Gaussian IID $(\sigma^2 = 1,\ N = 2,\ K = 16)$



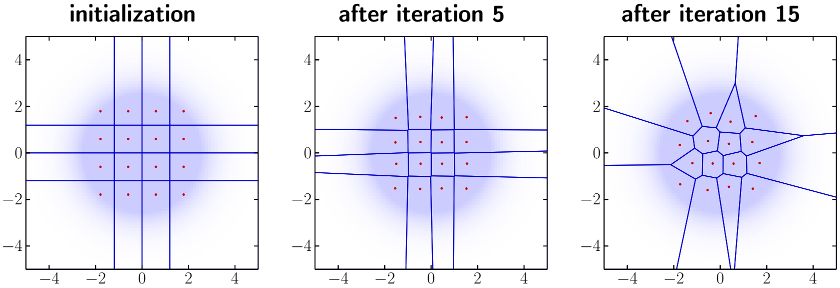


$$R = 2 \qquad\qquad R = 2$$

$$D = 0.122 \qquad\qquad D = 0.117$$

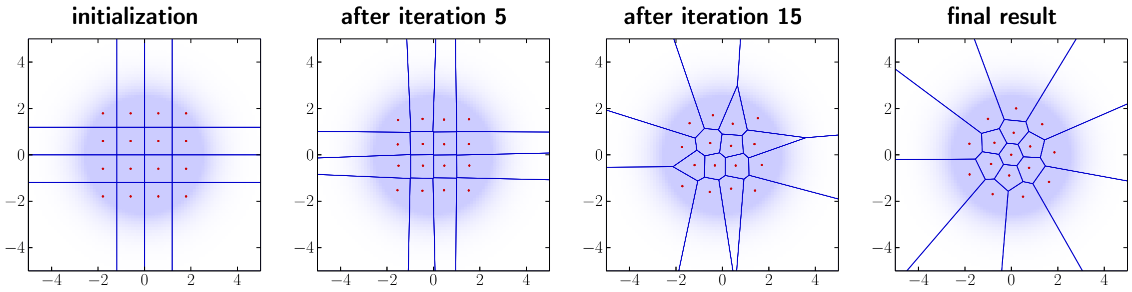
$$\text{SNR} = 9.12\,\text{dB} \qquad\qquad \text{SNR} = 9.31\,\text{dB}$$

Example: LBG Algorithm for Gaussian IID $(\sigma^2 = 1,\ N = 2,\ K = 16)$



initialization	after iteration 5	after iteration 15
$R = 2$	$R = 2$	$R = 2$
$D = 0.122$	$D = 0.117$	$D = 0.114$
$\text{SNR} = 9.12\,\text{dB}$	$\text{SNR} = 9.31\,\text{dB}$	$\text{SNR} = 9.43\,\text{dB}$

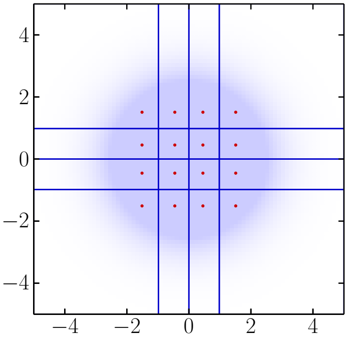
Example: LBG Algorithm for Gaussian IID $(\sigma^2 = 1,\ N = 2,\ K = 16)$



initialization	after iteration 5	after iteration 15	final result
$R = 2$	$R = 2$	$R = 2$	$R = 2$
$D = 0.122$	$D = 0.117$	$D = 0.114$	$D = 0.107$
$\text{SNR} = 9.12\,\text{dB}$	$\text{SNR} = 9.31\,\text{dB}$	$\text{SNR} = 9.43\,\text{dB}$	$\text{SNR} = 9.69\,\text{dB}$

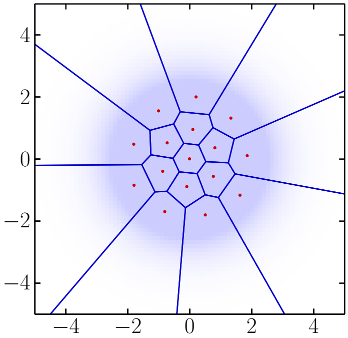
Comparison to Scalar Quantization: Gaussian IID $(\sigma^2 = 1,\ R = 2)$

Lloyd $(N = 1)$



$$D = 0.117$$
$$\text{SNR} = 9.30\,\text{dB}$$

LBG $(N = 2)$

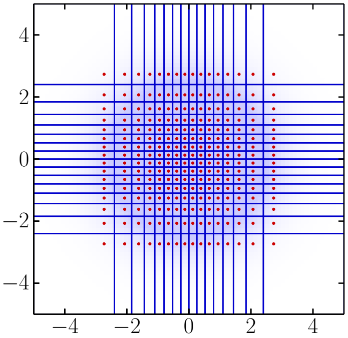


$$D = 0.107$$
$$\text{SNR} = 9.69\,\text{dB}$$

➡ **Improvement of 0.39 dB** (distortion reduction by factor 0.91)

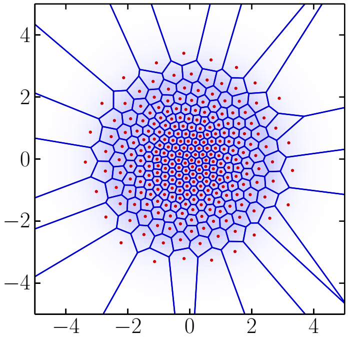
Comparison to Scalar Quantization: Gaussian IID $(\sigma^2 = 1,\ R = 4)$

Lloyd $(N = 1)$



$$D = 0.00951$$
$$\text{SNR} = 20.22\,\text{dB}$$

LBG $(N = 2)$

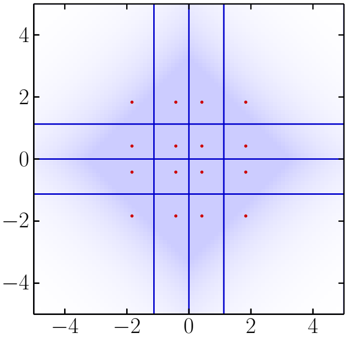


$$D = 0.00767$$
$$\text{SNR} = 21.15\,\text{dB}$$

➡ **Improvement of 0.93 dB** (distortion reduction by factor 0.81)

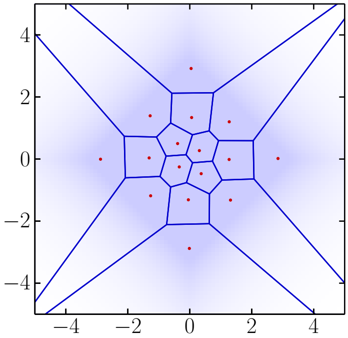
Comparison to Scalar Quantization: Laplacian IID $(\sigma^2 = 1,\ R = 2)$

Lloyd $(N = 1)$



$$D = 0.176$$
$$\text{SNR} = 7.54\,\text{dB}$$

LBG $(N = 2)$

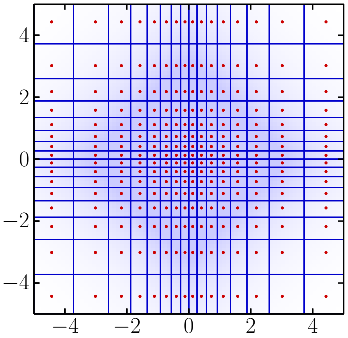


$$D = 0.129$$
$$\text{SNR} = 8.89\,\text{dB}$$

➡ **Improvement of 1.35 dB** (distortion reduction by factor 0.73)

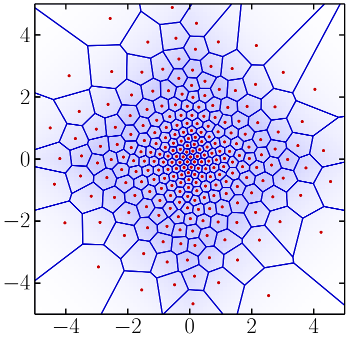
Comparison to Scalar Quantization: Laplacian IID $(\sigma^2 = 1,\ R = 4)$

Lloyd $(N = 1)$



$$D = 0.0153$$
$$\text{SNR} = 18.14\,\text{dB}$$

LBG $(N = 2)$



$$D = 0.0098$$
$$\text{SNR} = 20.08\,\text{dB}$$

➡ **Improvement of 1.94 dB** (distortion reduction by factor 0.64)

The Vector Quantizer Advantage

Gain over scalar quantization can be assigned to 3 effects:

The Vector Quantizer Advantage

Gain over scalar quantization can be assigned to 3 effects:

- **Space filling advantage**:
  - $\mathbb{Z}^N$ lattice is not most efficient sphere packing in N dimensions ($N > 1$)
  - Independent from source distribution or statistical dependencies
  - Maximum gain for $N \to \infty$: 1.53 dB

The Vector Quantizer Advantage

Gain over scalar quantization can be assigned to 3 effects:

- **Space filling advantage**:
  - $\mathbb{Z}^N$ lattice is not most efficient sphere packing in N dimensions ($N > 1$)
  - Independent from source distribution or statistical dependencies
  - Maximum gain for $N \rightarrow \infty$: 1.53 dB
- **Shape advantage**:
  - Exploit shape of source pdf
  - Can also be exploited using entropy-constrained scalar quantization

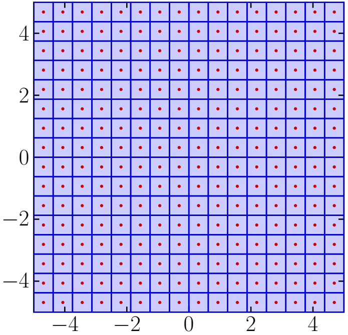
The Vector Quantizer Advantage

Gain over scalar quantization can be assigned to 3 effects:

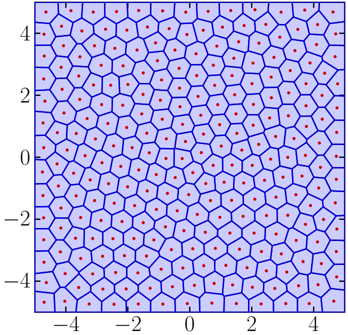
- **Space filling advantage**:
  - $\mathbb{Z}^N$ lattice is not most efficient sphere packing in N dimensions ($N > 1$)
  - Independent from source distribution or statistical dependencies
  - Maximum gain for $N \to \infty$: 1.53 dB
- **Shape advantage**:
  - Exploit shape of source pdf
  - Can also be exploited using entropy-constrained scalar quantization
- **Memory advantage**:
  - Exploit statistical dependencies of the source
  - Can also be exploited using predictive coding, transform coding, block entropy coding or conditional entropy coding

Space-Filling Advantage: LBG for Uniform IID Source

Lloyd ($N = 1$): SNR = 23.97 dB



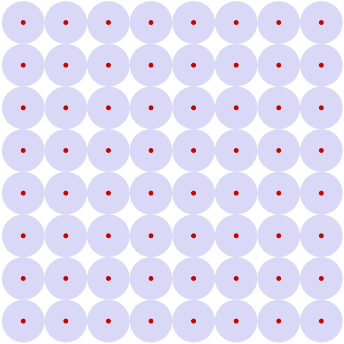
LBG ($N = 2$): SNR = 24.14 dB



- LBG algorithm approaches approximate hexagonal lattice

➡ Improvement of 0.17 dB

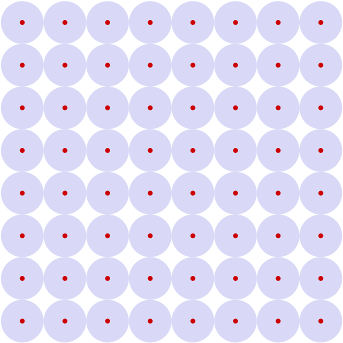
Space-Filling Advantage: Sphere Packing in N-dimensional Signal Space

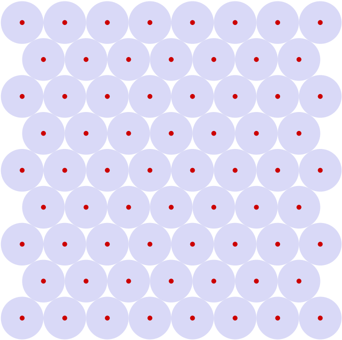


- Space filling gain: Densest packing of "optimal" quantization cells in signal space

➡ MSE distortion: **Densest packing of spheres in N-dimensional space**

Space-Filling Advantage: Sphere Packing in N-dimensional Signal Space

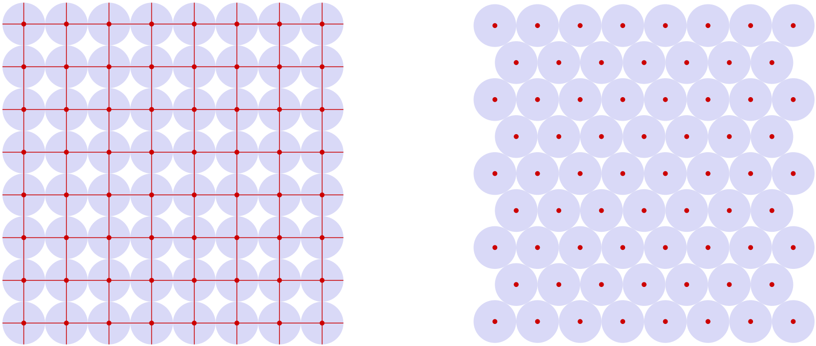




- Space filling gain: Densest packing of "optimal" quantization cells in signal space

→ MSE distortion: **Densest packing of spheres in N-dimensional space**

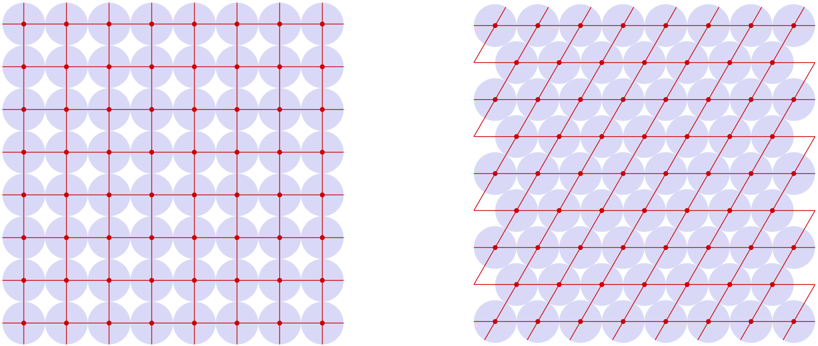
Space-Filling Advantage: Sphere Packing in N-dimensional Signal Space



- Space filling gain: Densest packing of "optimal" quantization cells in signal space

➡ MSE distortion: **Densest packing of spheres in N-dimensional space**

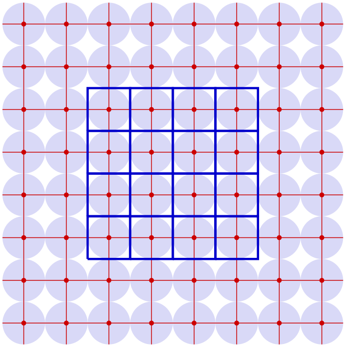
Space-Filling Advantage: Sphere Packing in N-dimensional Signal Space

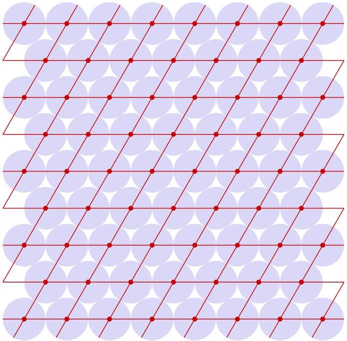


- Space filling gain: Densest packing of "optimal" quantization cells in signal space

→ MSE distortion: **Densest packing of spheres in N-dimensional space**

Space-Filling Advantage: Sphere Packing in N-dimensional Signal Space

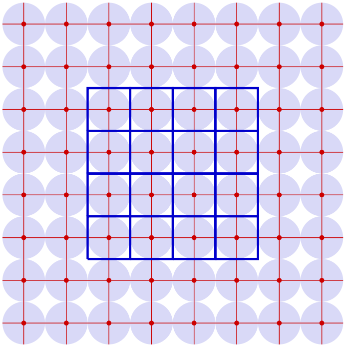


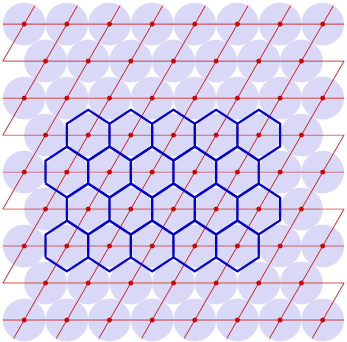


- Space filling gain: Densest packing of "optimal" quantization cells in signal space

➡ MSE distortion: **Densest packing of spheres in N-dimensional space**

Space-Filling Advantage: Sphere Packing in N-dimensional Signal Space

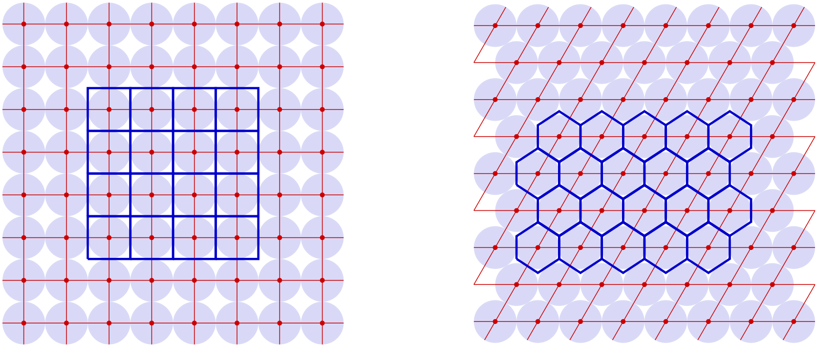




- Space filling gain: Densest packing of "optimal" quantization cells in signal space

➔ MSE distortion: **Densest packing of spheres in N-dimensional space**

Space-Filling Advantage: Sphere Packing in N-dimensional Signal Space

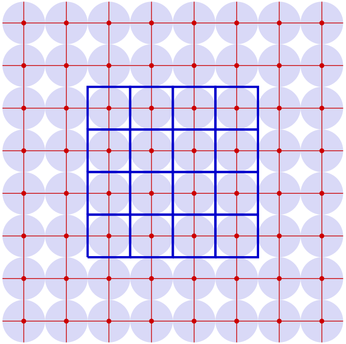


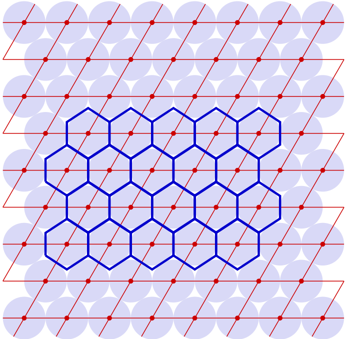
- Space filling gain: Densest packing of "optimal" quantization cells in signal space

➞ MSE distortion: **Densest packing of spheres in N-dimensional space**

  ➞ 2 dimensions: Hexagonal lattice (like honeycombs)

Space-Filling Advantage: Sphere Packing in N-dimensional Signal Space





- Space filling gain: Densest packing of "optimal" quantization cells in signal space

➡ MSE distortion: **Densest packing of spheres in N-dimensional space**

- ➡ 2 dimensions: Hexagonal lattice (like honeycombs)
- ➡ 3 dimensions: Cuboidal lattice (stapling of cannon balls / oranges)

Space-Filling Advantage: Sphere Packing Density

Center density

- Consider N-dimensional spheres with radius $r = 1$

Space-Filling Advantage: Sphere Packing Density

Center density

- Consider N-dimensional spheres with radius $r = 1$
- Measure for packing density: **Center density**

$$\delta = \frac{\text{average number of sphere centers}}{\text{unit volume}}$$

Space-Filling Advantage: Sphere Packing Density

Center density

- Consider N-dimensional spheres with radius $r = 1$
- Measure for packing density: **Center density**

$$\delta = \frac{\text{average number of sphere centers}}{\text{unit volume}}$$

- Example: $N = 1$ (SQ with intervals of size $2r = 2$)

$$\delta = \frac{1}{2}$$

Space-Filling Advantage: Sphere Packing Density

Center density

- Consider N-dimensional spheres with radius $r = 1$
- Measure for packing density: **Center density**

$$\delta = \frac{\text{average number of sphere centers}}{\text{unit volume}}$$

- Example: $N = 1$ (SQ with intervals of size $2r = 2$)

$$\delta = \frac{1}{2}$$

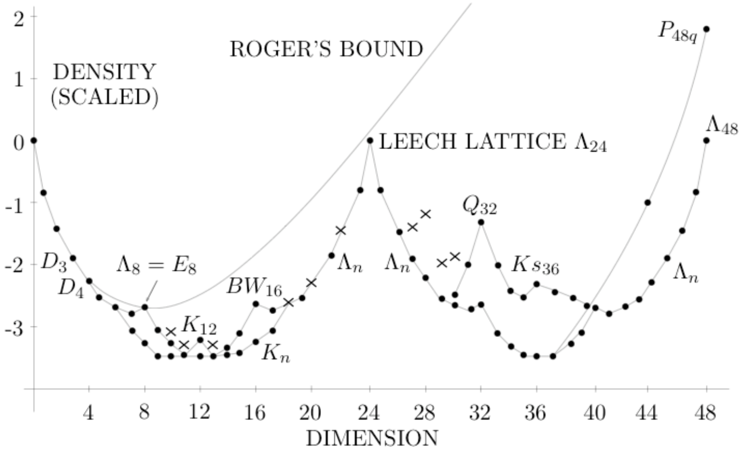
Roger's bound

- Theoretical upper bound for center density (last term being approximate)

$$\log_2 \delta \leq \frac{N}{2} \log_2 \left(\frac{N}{4e\pi} \right) + \frac{1}{2} \log_2 \left(\frac{\pi\, N^3}{e^2} \right) + \frac{21}{4N + 10}$$

Space-Filling Advantage: Densest Known Sphere Packings

- Densest known packings for dimensions $N \leq 48$ [Conway, Sloane, 1998]
- Vertical axis: $\log_2 \delta + N(24 - N)/96$



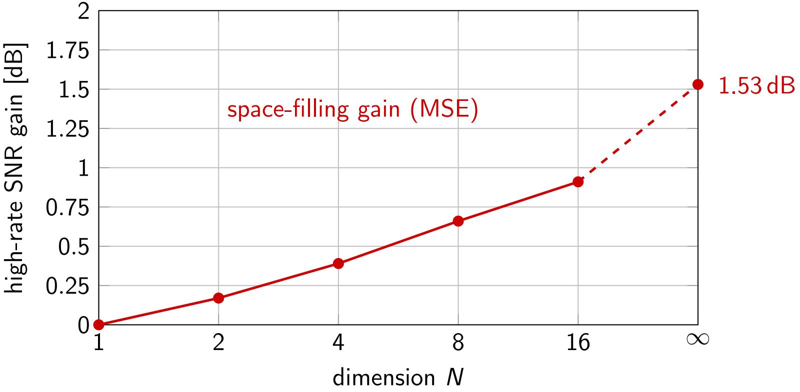
Space-Filling Advantage: Approximate SNR Gain

dimension	densest packing	(name)	highest kissing number	approximate gain [dB]
1	$\mathbb{Z}$	– Integer lattice	2	0
2	A_2	– Hexagonal lattice	6	**0.17**
3	$A_3 \simeq D_3$	– Cuboidal lattice	12	0.29
4	D_4		24	0.39
5	D_5		40	0.47
6	E_6		72	0.54
7	E_7		126	0.60
8	E_8	– Gosset lattice	240	0.66
9	Λ_9	– Laminated lattice	240	0.70
10	P_{10c}	– Non-lattice arrangement	336	0.74
12	K_{12}	– Coxeter-Todd lattice	756	0.81
16	$BW_{16} \simeq \Lambda_{16}$	– Barnes-Wall lattice	4320	0.91
24	Λ_{24}	– Leech lattice	196560	1.04
100				1.35
∞				**1.53**

Summary on Space-Filling Advantage

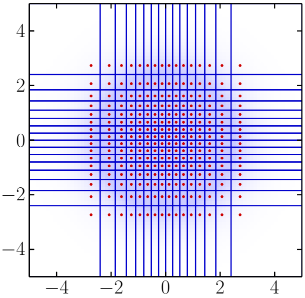
- Gain of unique to vector quantization: Packing of quantization cells in N-dimensional space
- Increases with quantizer dimension N

➡ Gain for $N \to \infty$: Difference between Shannon lower bound and ECSQ

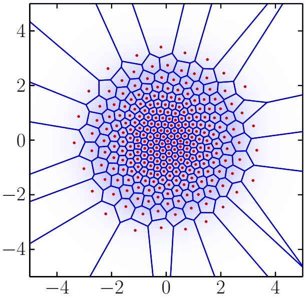


Shape Advantage: Gaussian IID $(\sigma^2 = 1,\ R = 4)$

Lloyd $(N = 1)$



LBG $(N = 2)$



$$\frac{D_{N=2}}{D_{N=1}} \approx 0.81$$

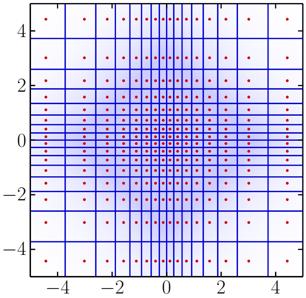
$$\Delta\mathrm{SNR} \approx 0.93\,\mathrm{dB}$$

Shape Advantage of Vector Quantizers

- Coding gain (0.93 dB for example) is larger than space-filling gain (0.17 dB for $N = 2$)
- Vector quantizer can better adapt to shape of pdf (even without entropy coding)

Shape Advantage: Laplacian IID $(\sigma^2 = 1,\ R = 4)$

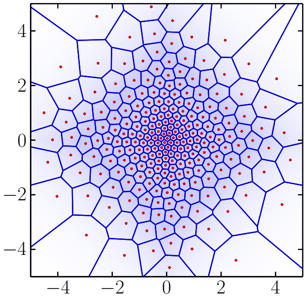
Lloyd $(N = 1)$



$$\frac{D_{N=2}}{D_{N=1}} \approx 0.64$$

$$\Delta\text{SNR} \approx 1.94\,\text{dB}$$

LBG $(N = 2)$



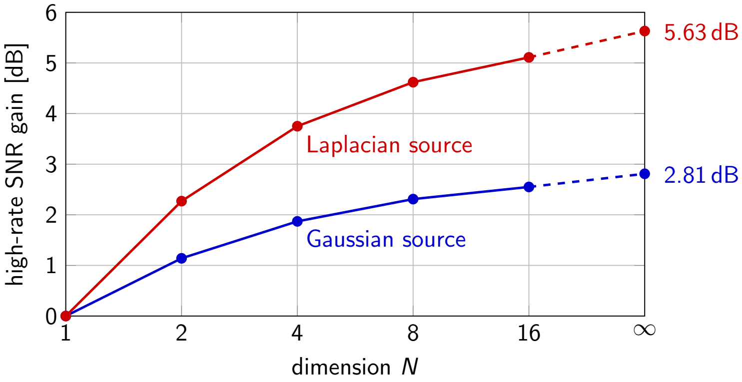
Shape Advantage of Vector Quantizers

- Coding gain (1.94 dB for example) is larger than space-filling gain (0.17 dB for $N = 2$)
- Vector quantizer can better adapt to shape of pdf (even without entropy coding)

Summary on Shape Advantage

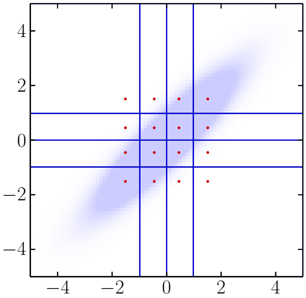
- Gain of VQ due to exploitation of shape of pdf (without entropy coding)
- Overall gain for iid source: Space-filling gain + shape gain

➡ Shape advantage can also be exploited by entropy-constrained scalar quantization



Memory Advantage: Gauss-Markov $(\sigma^2 = 1,\ \varrho = 0.9,\ R = 2)$

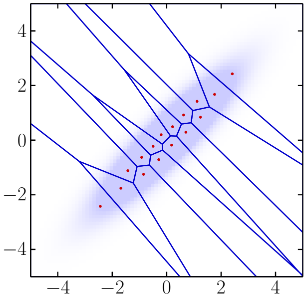
Lloyd $(N = 1)$



$$\frac{D_{N=2}}{D_{N=1}} \approx 0.38$$

$$\Delta\mathrm{SNR} \approx 4.20\,\mathrm{dB}$$

LBG $(N = 2)$

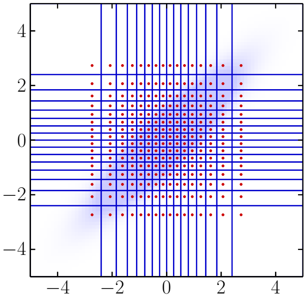


Memory Advantage of Vector Quantizers

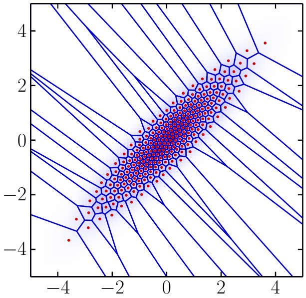
- Large coding gain (4.20 dB for example) for sources with memory
- Vector quantizer can exploit dependencies between samples

Memory Advantage: Gauss-Markov $(\sigma^2 = 1,\ \varrho = 0.9,\ R = 4)$

Lloyd $(N = 1)$



LBG $(N = 2)$



$$\frac{D_{N=2}}{D_{N=1}} \approx 0.35$$

$$\Delta\text{SNR} \approx 4.55\,\text{dB}$$

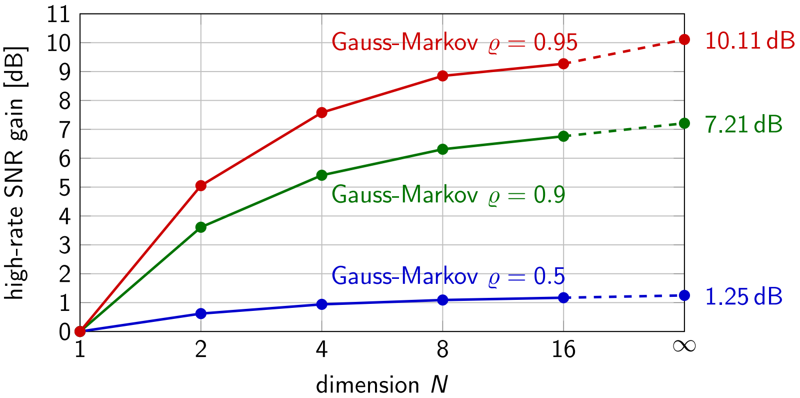
Memory Advantage of Vector Quantizers

- Large coding gain (4.55 dB for example) for sources with memory
- Vector quantizer can exploit dependencies between samples

Summary on Memory Advantage

- Gain of VQ due to exploitation of dependencies between samples
- Largest gain to be made for sources with strong statistical dependencies

→ Exploitation of memory advantage is one of the most relevant aspects in source coding



Optimal Vector Quantizer for Variable-Length Coding

Optimal Vector Quantizer with Consideration of Entropy Coding

- Similar to Scalar Entropy-Constrained Lloyd Quantizer

Optimal Vector Quantizer for Variable-Length Coding

Optimal Vector Quantizer with Consideration of Entropy Coding

- Similar to Scalar Entropy-Constrained Lloyd Quantizer

➡ Minimization of Lagrangian cost for given Lagrange multiplier λ

$$\begin{aligned} J &= D + \lambda \cdot R \\ &= \frac{1}{N} \sum_{\forall k} \int_{\mathcal{C}_k} \left|\left| \boldsymbol{s} - \boldsymbol{s}'_{\boldsymbol{k}} \right|\right|_2^2 f(\boldsymbol{s}) \, \mathrm{d}\boldsymbol{s} \;+\; \frac{\lambda}{N} \sum_{\forall k} \ell_k \int_{\mathcal{C}_k} f(\boldsymbol{s}) \, \mathrm{d}\boldsymbol{s} \end{aligned}$$

Optimal Vector Quantizer for Variable-Length Coding

Optimal Vector Quantizer with Consideration of Entropy Coding

- Similar to Scalar Entropy-Constrained Lloyd Quantizer

➡ Minimization of Lagrangian cost for given Lagrange multiplier λ

$$\begin{aligned} J &= D + \lambda \cdot R \\ &= \frac{1}{N} \sum_{\forall k} \int_{\mathcal{C}_k} \left|\left| \boldsymbol{s} - \boldsymbol{s}'_k \right|\right|_2^2 f(\boldsymbol{s}) \, \mathrm{d}\boldsymbol{s} \;+\; \frac{\lambda}{N} \sum_{\forall k} \ell_k \int_{\mathcal{C}_k} f(\boldsymbol{s}) \, \mathrm{d}\boldsymbol{s} \end{aligned}$$

- Lagrange multiplier $\lambda > 0$ determines operation point (trade-off between rate and distortion)

Optimal Vector Quantizer for Variable-Length Coding

Optimal Vector Quantizer with Consideration of Entropy Coding

- Similar to Scalar Entropy-Constrained Lloyd Quantizer

➡ Minimization of Lagrangian cost for given Lagrange multiplier λ

$$
\begin{aligned}
J &= D + \lambda \cdot R \\
&= \frac{1}{N} \sum_{\forall k} \int_{\mathcal{C}_k} ||\boldsymbol{s} - \boldsymbol{s}'_k||_2^2 \, f(\boldsymbol{s}) \, \mathrm{d}\boldsymbol{s} + \frac{\lambda}{N} \sum_{\forall k} \ell_k \int_{\mathcal{C}_k} f(\boldsymbol{s}) \, \mathrm{d}\boldsymbol{s}
\end{aligned}
$$

- Lagrange multiplier $\lambda > 0$ determines operation point (trade-off between rate and distortion)

Optimize Quantizer for given Lagrange multiplier

- Derive necessary conditions for optimality (similar to EC Lloyd quantizer)
- Construct iterative algorithm for designing quantizer
- Similar as for EC Lloyd: Use large number of intervals in initialization

Optimality Conditions for Variable-Length Coding

Necessary Conditions for Optimality (MSE distortion)

1. Centroid condition (for reconstruction vectors $\boldsymbol{s}'_k$, same as for LBG)

$$\boxed{\boldsymbol{s}'_k = \mathrm{E}\{\, \boldsymbol{S} \,|\, \boldsymbol{S} \in \mathcal{C}_k \,\} = \frac{1}{p_k} \int_{\mathcal{C}_k} \boldsymbol{s}\, f(\boldsymbol{s})\, \mathrm{d}\boldsymbol{s}}$$

(training set: take average of assigned vectors)

Optimality Conditions for Variable-Length Coding

Necessary Conditions for Optimality (MSE distortion)

1. Centroid condition (for reconstruction vectors $\boldsymbol{s}'_k$, same as for LBG)

$$\boxed{\boldsymbol{s}'_k = \mathrm{E}\{\,\boldsymbol{S} \,|\, \boldsymbol{S} \in \mathcal{C}_k\,\} = \frac{1}{p_k}\int_{\mathcal{C}_k} \boldsymbol{s}\, f(\boldsymbol{s})\, \mathrm{d}\boldsymbol{s}}$$

(training set: take average of assigned vectors)

2. Entropy condition (for codeword length ℓ_k, same as for EC Llloyd)

$$\boxed{\ell_k = -\log_2 p_k = -\log_2 \int_{\mathcal{C}_k} f(\boldsymbol{s})\, \mathrm{d}\boldsymbol{s}}$$

(training set: count assigned vectors)

Optimality Conditions for Variable-Length Coding

Necessary Conditions for Optimality (MSE distortion)

1. Centroid condition (for reconstruction vectors $\boldsymbol{s}'_k$, same as for LBG)

$$\boxed{\boldsymbol{s}'_k = \mathrm{E}\{\, \boldsymbol{S} \,|\, \boldsymbol{S} \in \mathcal{C}_k \,\} = \frac{1}{p_k} \int_{\mathcal{C}_k} \boldsymbol{s}\, f(\boldsymbol{s})\, \mathrm{d}\boldsymbol{s}}$$

(training set: take average of assigned vectors)

2. Entropy condition (for codeword length ℓ_k, same as for EC Llloyd)

$$\boxed{\ell_k = -\log_2 p_k = -\log_2 \int_{\mathcal{C}_k} f(\boldsymbol{s})\, \mathrm{d}\boldsymbol{s}}$$

(training set: count assigned vectors)

3. Modified nearest neighbour condition (for quantization cells $\mathcal{C}_k$ / encoder mapping $\alpha(.)$)

$$\boxed{\alpha(\boldsymbol{s}) = \arg\min_{\forall k} \big|\big|\boldsymbol{s} - \boldsymbol{s}'_k\big|\big|_2^2 + \lambda \cdot \ell_k}$$

The Chou-Lookabough-Gray (CLG) Algorithm for a Training Set (MSE)

Given is:

- the Lagrange multiplier $\lambda > 0$
- a sufficiently large realization $\{\boldsymbol{s}_n\}$ of considered source

The Chou-Lookabough-Gray (CLG) Algorithm for a Training Set (MSE)

Given is:

- the Lagrange multiplier $\lambda > 0$
- a sufficiently large realization $\{\boldsymbol{s}_n\}$ of considered source

Iterative quantizer design

1. Choose an initial set of K reconstruction vectors $\{\boldsymbol{s}'_k\}$ and codeword length $\{\ell_k\}$

The Chou-Lookabaugh-Gray (CLG) Algorithm for a Training Set (MSE)

Given is:

- the Lagrange multiplier $\lambda > 0$
- a sufficiently large realization $\{\boldsymbol{s}_n\}$ of considered source

Iterative quantizer design

1. Choose an initial set of K reconstruction vectors $\{\boldsymbol{s}'_k\}$ and codeword length $\{\ell_k\}$
2. Associate all vectors of the training set $\{\boldsymbol{s}_n\}$ with one of the quantization cells $\mathcal{C}_k$

$$q(\boldsymbol{s}_n) = \arg\min_{\forall k} \left|\left|\boldsymbol{s}_n - \boldsymbol{s}'_k\right|\right|_2^2 + \lambda \cdot \ell_k \qquad \text{(modified nearest neighbor condition)}$$

The Chou-Lookabaugh-Gray (CLG) Algorithm for a Training Set (MSE)

Given is:
- the Lagrange multiplier $\lambda > 0$
- a sufficiently large realization $\{\boldsymbol{s}_n\}$ of considered source

Iterative quantizer design

1. Choose an initial set of K reconstruction vectors $\{\boldsymbol{s}'_k\}$ and codeword length $\{\ell_k\}$
2. Associate all vectors of the training set $\{\boldsymbol{s}_n\}$ with one of the quantization cells $\mathcal{C}_k$

$$q(\boldsymbol{s}_n) = \arg\min_{\forall k} \left|\left|\boldsymbol{s}_n - \boldsymbol{s}'_k\right|\right|_2^2 + \lambda \cdot \ell_k \qquad \text{(modified nearest neighbor condition)}$$

3. Update the reconstruction vectors $\{\boldsymbol{s}'_k\}$ and codeword length $\{\ell_k\}$ according to

$$\boldsymbol{s}'_k = \frac{1}{n(k)} \sum_{\forall n:\, q(\boldsymbol{s}_n)=k} \boldsymbol{s}_n \qquad \text{and} \qquad \ell_k = -\log_2\left(\frac{n(k)}{\sum_{\forall i} n(i)}\right)$$

where $n(k)$ is the number of sample vectors $\boldsymbol{s}_n$ assigned to $\mathcal{C}_k$

The Chou-Lookabough-Gray (CLG) Algorithm for a Training Set (MSE)

Given is:
- the Lagrange multiplier $\lambda > 0$
- a sufficiently large realization $\{\boldsymbol{s}_n\}$ of considered source

Iterative quantizer design

1. Choose an initial set of K reconstruction vectors $\{\boldsymbol{s}'_k\}$ and codeword length $\{\ell_k\}$
2. Associate all vectors of the training set $\{\boldsymbol{s}_n\}$ with one of the quantization cells $\mathcal{C}_k$

$$q(\boldsymbol{s}_n) = \arg\min_{\forall k} \; \left|\left|\boldsymbol{s}_n - \boldsymbol{s}'_k\right|\right|_2^2 + \lambda \cdot \ell_k \qquad \text{(modified nearest neighbor condition)}$$

3. Update the reconstruction vectors $\{\boldsymbol{s}'_k\}$ and codeword length $\{\ell_k\}$ according to

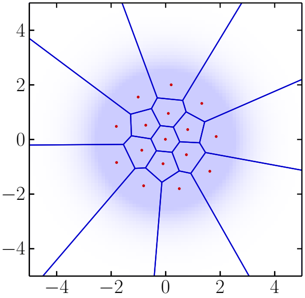
$$\boldsymbol{s}'_k = \frac{1}{n(k)} \sum_{\forall n:\, q(\boldsymbol{s}_n)=k} \boldsymbol{s}_n \qquad \text{and} \qquad \ell_k = -\log_2\left(\frac{n(k)}{\sum_{\forall i} n(i)}\right)$$

   where $n(k)$ is the number of sample vectors $\boldsymbol{s}_n$ assigned to $\mathcal{C}_k$

4. Repeat the previous two steps until convergence

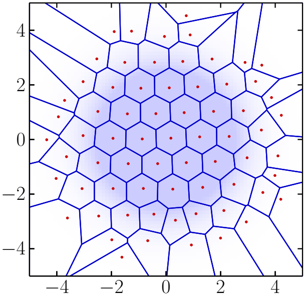
Entropy-Constrained Vector Quantizer: Gaussian IID ($\sigma^2 = 1,\ R = 2$)

LBG ($N = 2$)



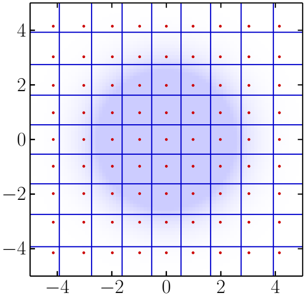
$$D = 0.107$$
$$\text{SNR} = 9.69\,\text{dB}$$

CLG ($N = 2$)



$$D = 0.086$$
$$\text{SNR} = 10.68\,\text{dB}$$

EC Lloyd ($N = 1$)



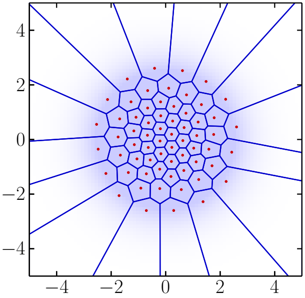
$$D = 0.089$$
$$\text{SNR} = 10.51\,\text{dB}$$

➡ Large gain (1.0 dB) relative to LBG algorithm (fixed-length coding)

➡ Gain relative to EC Lloyd reduces to space-filling gain (0.17 dB for $N = 2$)

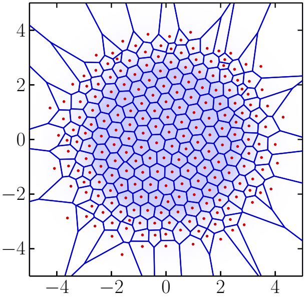
Entropy-Constrained Vector Quantizer: Gaussian IID ($\sigma^2 = 1$, $R = 3$)

LBG ($N = 2$)



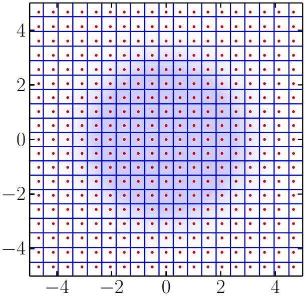
$$D = 0.0296$$
$$\text{SNR} = 15.29\,\text{dB}$$

CLG ($N = 2$)



$$D = 0.0214$$
$$\text{SNR} = 16.70\,\text{dB}$$

EC Lloyd ($N = 1$)



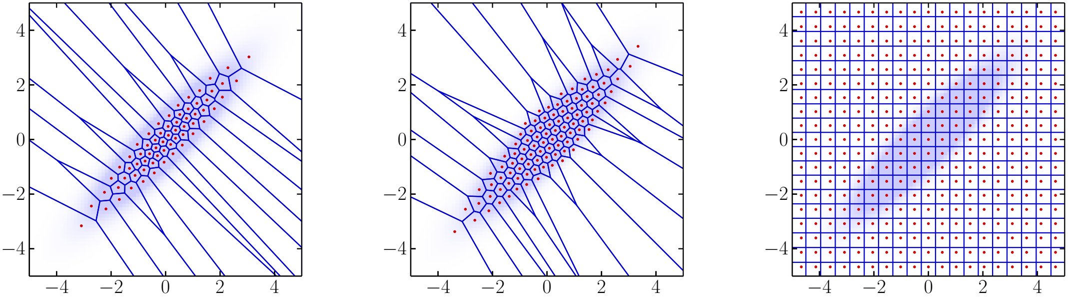
$$D = 0.0222$$
$$\text{SNR} = 16.53\,\text{dB}$$

➡ Large gain (1.4 dB) relative to LBG algorithm (fixed-length coding)

➡ Gain relative to EC Lloyd reduces to space-filling gain (0.17 dB for $N = 2$)

Entropy-Constrained Vector Quantizer: Gauss-Markov ($\sigma^2 = 1$, $\varrho = 0.9$, $R = 3$)

LBG ($N = 2$) | **CLG** ($N = 2$) | **EC Lloyd** ($N = 1$)



$D = 0.0125$	$D = 0.0099$	$D = 0.0222$
$\text{SNR} = 19.04\,\text{dB}$	$\text{SNR} = 20.06\,\text{dB}$	$\text{SNR} = 16.53\,\text{dB}$

➡ Large gain (1.0 dB) relative to LBG algorithm (fixed-length coding)

➡ Gain relative to EC Lloyd: Sum of memory gain and space-filling gain

Coding Efficiency of Vector Quantizers

Vector Quantizer Advantages

- Space-filling advantage
  - ➡ Unique to vector quantization: (0.17 dB for $N = 2$; 1.53 dB for $N \to \infty$)

Coding Efficiency of Vector Quantizers

Vector Quantizer Advantages

- Space-filling advantage
  - ➡ Unique to vector quantization: (0.17 dB for $N = 2$; 1.53 dB for $N \rightarrow \infty$)
- Shape advantage
  - ➡ Can also be exploited by entropy-constrained scalar quantization

Coding Efficiency of Vector Quantizers

Vector Quantizer Advantages

- Space-filling advantage
  - ➡ Unique to vector quantization: (0.17 dB for $N = 2$; 1.53 dB for $N \to \infty$)
- Shape advantage
  - ➡ Can also be exploited by entropy-constrained scalar quantization
- Memory advantage
  - ➡ Can also be (partly) exploited by other coding techniques (topic of next lectures)

Coding Efficiency of Vector Quantizers

Vector Quantizer Advantages

- Space-filling advantage
  - ➜ Unique to vector quantization: ($0.17\,\mathrm{dB}$ for $N = 2$; $1.53\,\mathrm{dB}$ for $N \to \infty$)
- Shape advantage
  - ➜ Can also be exploited by entropy-constrained scalar quantization
- Memory advantage
  - ➜ Can also be (partly) exploited by other coding techniques (topic of next lectures)

Coding Efficiency of Vector Quantizers

- Optimal vector quantizers provide coding efficiency gains relative to scalar quantizers

Coding Efficiency of Vector Quantizers

Vector Quantizer Advantages

- Space-filling advantage
  - ➜ Unique to vector quantization: (0.17 dB for $N = 2$; 1.53 dB for $N \to \infty$)
- Shape advantage
  - ➜ Can also be exploited by entropy-constrained scalar quantization
- Memory advantage
  - ➜ Can also be (partly) exploited by other coding techniques (topic of next lectures)

Coding Efficiency of Vector Quantizers

- Optimal vector quantizers provide coding efficiency gains relative to scalar quantizers
- IID sources: Only space-filling gain (when comparing entropy-constrained designs)

Coding Efficiency of Vector Quantizers

Vector Quantizer Advantages

- Space-filling advantage
  - ➡ Unique to vector quantization: (0.17 dB for $N = 2$; 1.53 dB for $N \to \infty$)
- Shape advantage
  - ➡ Can also be exploited by entropy-constrained scalar quantization
- Memory advantage
  - ➡ Can also be (partly) exploited by other coding techniques (topic of next lectures)

Coding Efficiency of Vector Quantizers

- Optimal vector quantizers provide coding efficiency gains relative to scalar quantizers
- IID sources: Only space-filling gain (when comparing entropy-constrained designs)
- Sources with memory: Most important aspect is the memory advantage

Coding Efficiency of Vector Quantizers

Vector Quantizer Advantages

- Space-filling advantage
  - ➡ Unique to vector quantization: (0.17 dB for $N = 2$; 1.53 dB for $N \to \infty$)
- Shape advantage
  - ➡ Can also be exploited by entropy-constrained scalar quantization
- Memory advantage
  - ➡ Can also be (partly) exploited by other coding techniques (topic of next lectures)

Coding Efficiency of Vector Quantizers

- Optimal vector quantizers provide coding efficiency gains relative to scalar quantizers
- IID sources: Only space-filling gain (when comparing entropy-constrained designs)
- Sources with memory: Most important aspect is the memory advantage

➡ **Vector quantizers can asymptotically achieve rate-distortion bound for $N \to \infty$**

Complexity of Vector Quantization

Decoding Complexity

- In principle: Table look-up (using transmitted quantization indexes)
- Extremely large memory requirements for large N

Complexity of Vector Quantization

Decoding Complexity

- In principle: Table look-up (using transmitted quantization indexes)
- Extremely large memory requirements for large N

Encoding Complexity

- Finding the "closest reconstruction vector" can become very complex
- Designing a good vector quantizer is already very complex

Complexity of Vector Quantization

Decoding Complexity

- In principle: Table look-up (using transmitted quantization indexes)
- Extremely large memory requirements for large N

Encoding Complexity

- Finding the "closest reconstruction vector" can become very complex
- Designing a good vector quantizer is already very complex

Usage of Vector Quantization

- Unconstrained vector quantizers are rarely used in practice

Complexity of Vector Quantization

Decoding Complexity

- In principle: Table look-up (using transmitted quantization indexes)
- Extremely large memory requirements for large N

Encoding Complexity

- Finding the "closest reconstruction vector" can become very complex
- Designing a good vector quantizer is already very complex

Usage of Vector Quantization

- Unconstrained vector quantizers are rarely used in practice

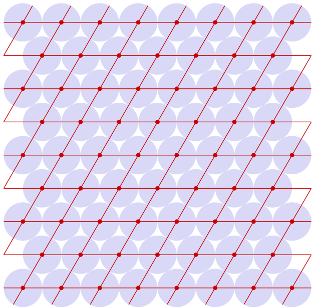
➞ Reduce complexity by imposing structural constraints
  - Tree-structured vector quantizers
  - Gain-shape vector quantizers
  - Lattice vector quantizers (important special case: Transform coding)
  - Trellis-coded quantization

Lattice Vector Quantizers & Transform Coding

Lattice Vector Quantizer

- Reconstruction vectors are located on multi-dimensional lattice
  - Lattice is specified by N "basis vectors" $\{\boldsymbol{b}_k\}$
  - Reconstruction vectors given by matrix of "basis vectors"

$$\boldsymbol{s}'_{k_1,k_2,\cdots,k_N} = \boldsymbol{M} \cdot [k_1, k_2, \cdots, k_N]^{\mathrm{T}}$$



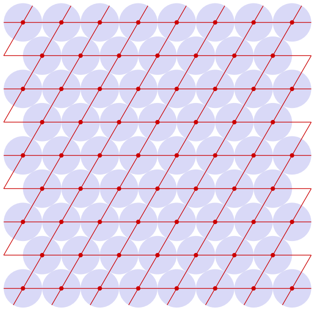
Lattice Vector Quantizers & Transform Coding

Lattice Vector Quantizer

- Reconstruction vectors are located on multi-dimensional lattice
  - Lattice is specified by N “basis vectors” $\{\boldsymbol{b}_k\}$
  - Reconstruction vectors given by matrix of “basis vectors”

$$\boldsymbol{s}'_{k_1,k_2,\cdots,k_N} = \boldsymbol{M} \cdot [k_1, k_2, \cdots, k_N]^{\mathrm{T}}$$

- Simple decoder operation possible



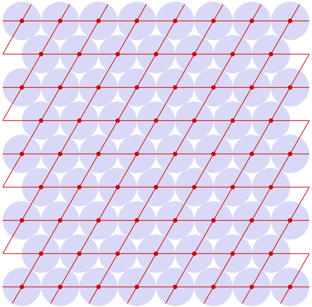
Lattice Vector Quantizers & Transform Coding

Lattice Vector Quantizer

- Reconstruction vectors are located on multi-dimensional lattice
  - Lattice is specified by N "basis vectors" $\{\boldsymbol{b}_k\}$
  - Reconstruction vectors given by matrix of "basis vectors"

$$\boldsymbol{s}'_{k_1,k_2,\cdots,k_N} = \boldsymbol{M} \cdot [k_1, k_2, \cdots, k_N]^{\mathrm{T}}$$

- Simple decoder operation possible
- Less complex encoding (can still by very complex for large N)



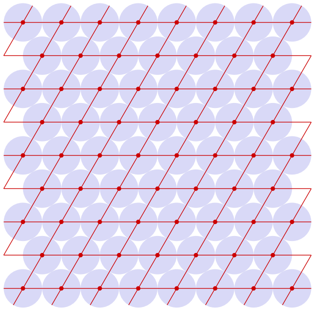
Lattice Vector Quantizers & Transform Coding

Lattice Vector Quantizer

- Reconstruction vectors are located on multi-dimensional lattice
  - Lattice is specified by N “basis vectors” $\{\boldsymbol{b}_k\}$
  - Reconstruction vectors given by matrix of “basis vectors”

$$\boldsymbol{s}'_{k_1,k_2,\cdots,k_N} = \boldsymbol{M} \cdot [k_1, k_2, \cdots, k_N]^{\mathrm{T}}$$

- Simple decoder operation possible
- Less complex encoding (can still by very complex for large N)



Transform Coding

- Lattice vector quantizer with orthonormal “basis vectors”
- Very simple encoding and decoding

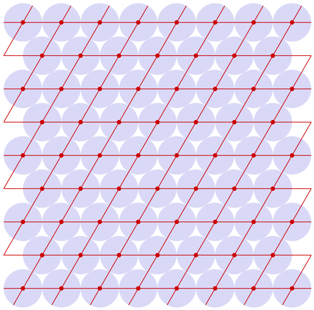
Lattice Vector Quantizers & Transform Coding

Lattice Vector Quantizer

- Reconstruction vectors are located on multi-dimensional lattice
  - Lattice is specified by N “basis vectors” $\{\boldsymbol{b}_k\}$
  - Reconstruction vectors given by matrix of “basis vectors”

$$\boldsymbol{s}'_{k_1,k_2,\cdots,k_N} = \boldsymbol{M} \cdot [k_1, k_2, \cdots, k_N]^{\mathrm{T}}$$

- Simple decoder operation possible
- Less complex encoding (can still by very complex for large N)



Transform Coding

- Lattice vector quantizer with orthonormal “basis vectors”
- Very simple encoding and decoding

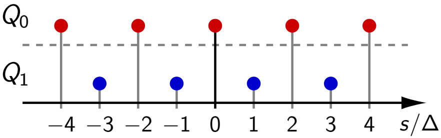
➡ One of the most often used approaches in lossy coding

➡ Will discuss in detail in next lectures

Trellis-Coded Quantization (TCQ)

Quantizer Design & Decoding Process

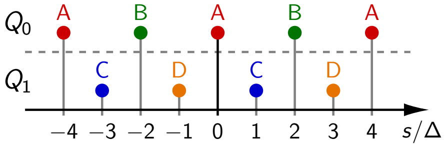
- Two scalar quantizers



Trellis-Coded Quantization (TCQ)

Quantizer Design & Decoding Process

- Two scalar quantizers + Prodecure for switching between quantizers (state machine with 2^N states)

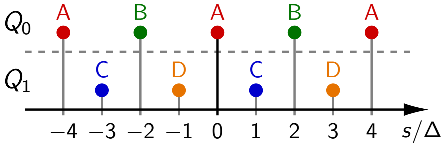


state	quantizer	next state
0	Q_0	(A,B) $\mapsto$ (0,1)
1	Q_1	(C,D) $\mapsto$ (2,3)
2	Q_0	(A,B) $\mapsto$ (1,0)
3	Q_1	(C,D) $\mapsto$ (3,2)

Trellis-Coded Quantization (TCQ)

Quantizer Design & Decoding Process

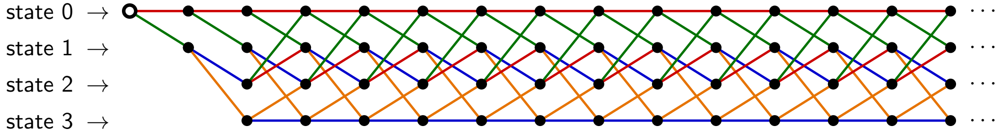
- Two scalar quantizers + Prodecure for switching between quantizers (state machine with 2^N states)



state	quantizer	next state
0	Q_0	(A,B) $\mapsto$ (0,1)
1	Q_1	(C,D) $\mapsto$ (2,3)
2	Q_0	(A,B) $\mapsto$ (1,0)
3	Q_1	(C,D) $\mapsto$ (3,2)

Encoding Process

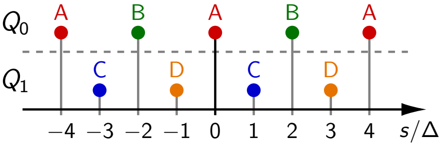
- Trellis formulation of possible quantizer switching



Trellis-Coded Quantization (TCQ)

Quantizer Design & Decoding Process

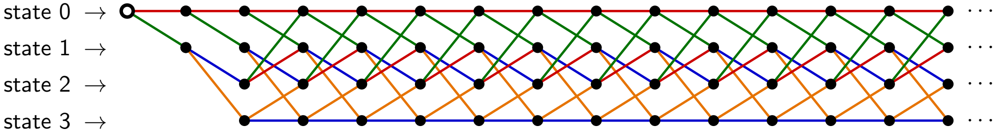
- Two scalar quantizers + Prodecure for switching between quantizers (state machine with 2^N states)



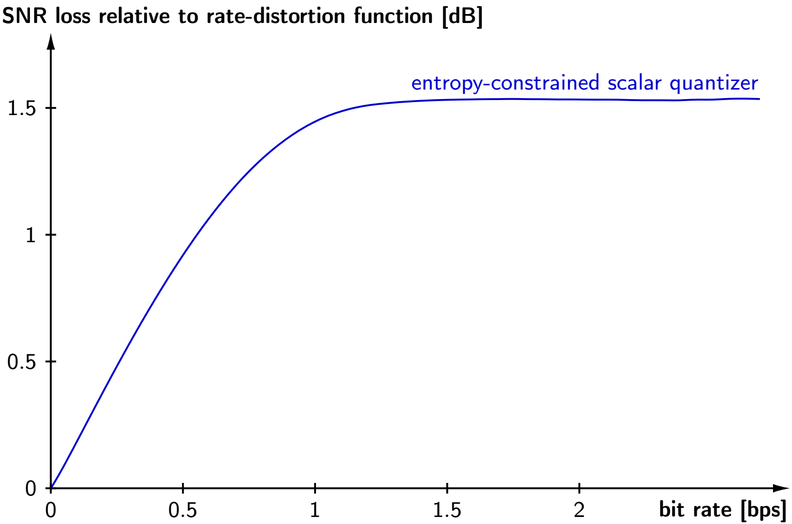
state	quantizer	next state
0	Q_0	(A,B) $\mapsto$ (0,1)
1	Q_1	(C,D) $\mapsto$ (2,3)
2	Q_0	(A,B) $\mapsto$ (1,0)
3	Q_1	(C,D) $\mapsto$ (3,2)

Encoding Process

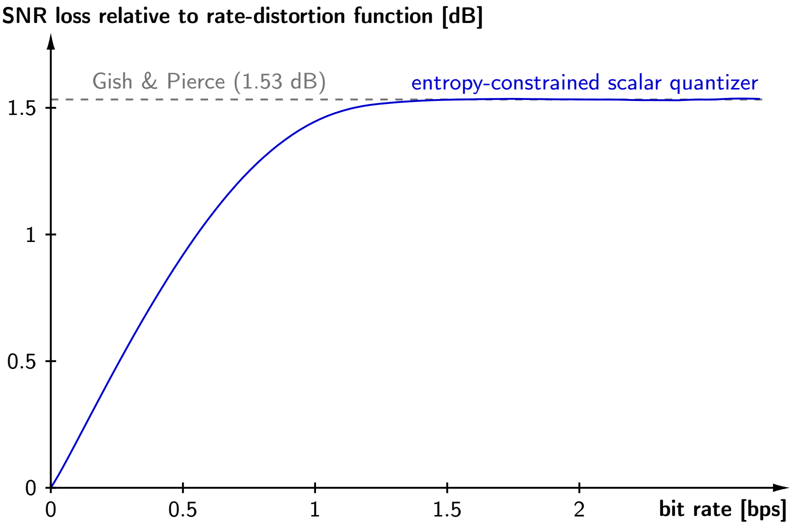
- Trellis formulation of possible quantizer switching ➡ **Viterbi algorithm**



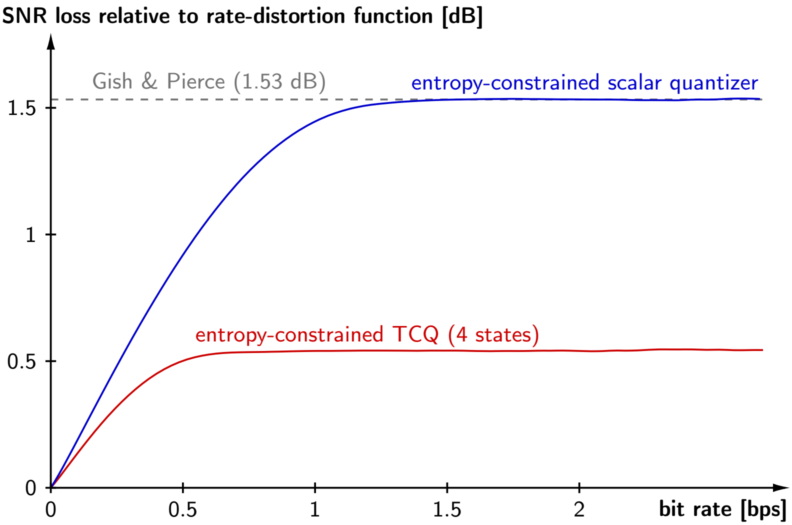
Example: TCQ Performance for Gaussian IID



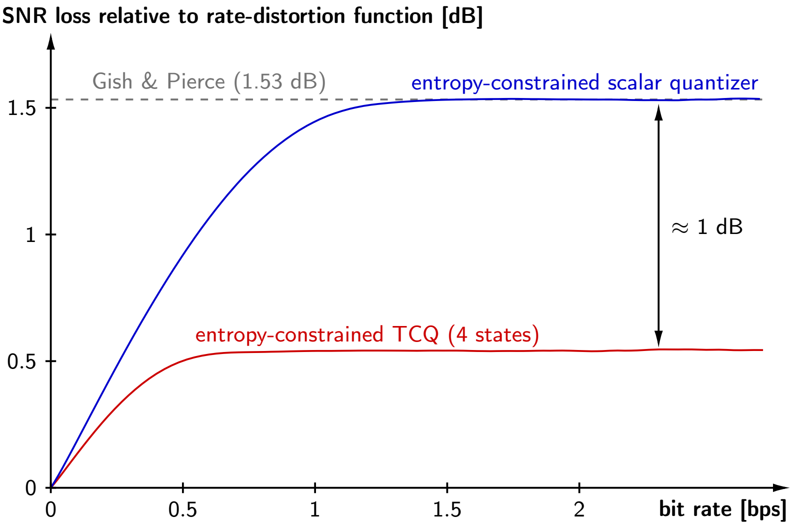
Example: TCQ Performance for Gaussian IID



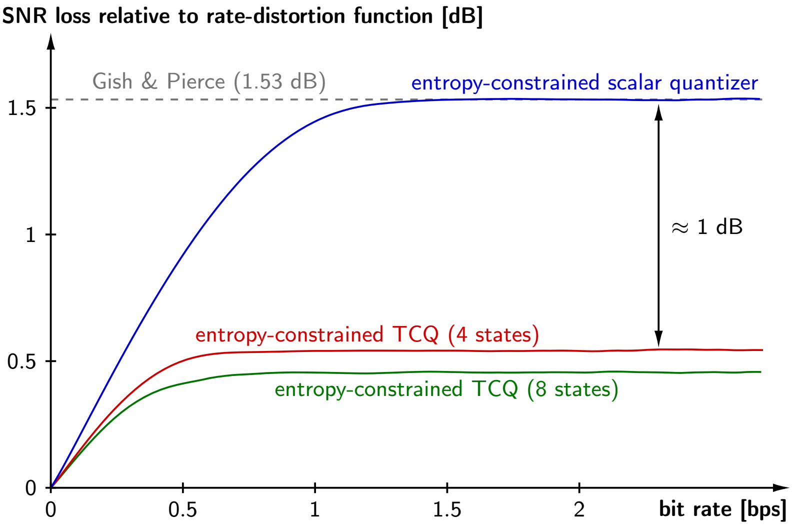
Example: TCQ Performance for Gaussian IID



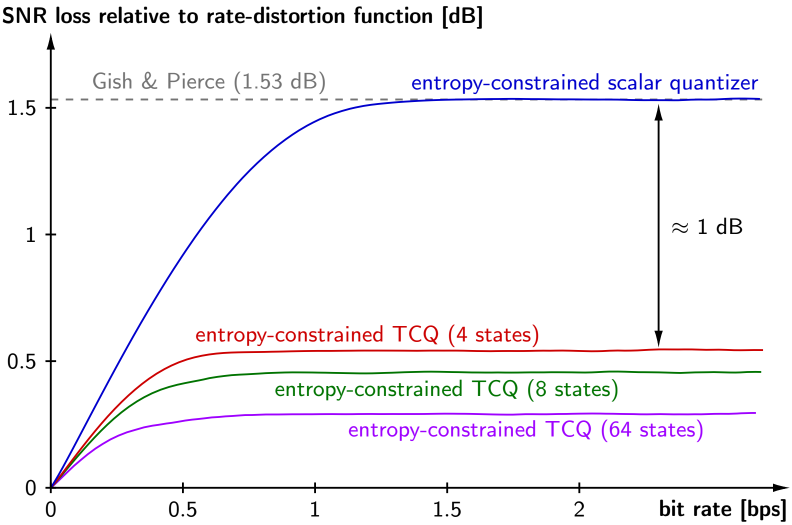
Example: TCQ Performance for Gaussian IID



Example: TCQ Performance for Gaussian IID



Example: TCQ Performance for Gaussian IID



Summary of Lecture

Vector Quantization (VQ)

- Straightforward extension of scalar quantization to higher dimensions N
- Opt. VQ with fixed-length codes: Similar to Lloyd quantizer
- Opt. VQ with variable-length codes: Similar to EC-Lloyd quantizer

Summary of Lecture

Vector Quantization (VQ)

- Straightforward extension of scalar quantization to higher dimensions N
- Opt. VQ with fixed-length codes: Similar to Lloyd quantizer
- Opt. VQ with variable-length codes: Similar to EC-Lloyd quantizer

Vector Quantizer Advantages

- Space-filling advantage: Unique to vector quantizers (1.53 dB for $N \to \infty$)
- Shape advantage: Can also be exploited by ECSQ
- Memory advantage: Can also be exploited by other coding techniques

Summary of Lecture

Vector Quantization (VQ)

- Straightforward extension of scalar quantization to higher dimensions N
- Opt. VQ with fixed-length codes: Similar to Lloyd quantizer
- Opt. VQ with variable-length codes: Similar to EC-Lloyd quantizer

Vector Quantizer Advantages

- Space-filling advantage: Unique to vector quantizers (1.53 dB for $N \to \infty$)
- Shape advantage: Can also be exploited by ECSQ
- Memory advantage: Can also be exploited by other coding techniques

Vector Quantization can achieve Rate-Distortion Bound! – Are we done?

Summary of Lecture

Vector Quantization (VQ)

- Straightforward extension of scalar quantization to higher dimensions N
- Opt. VQ with fixed-length codes: Similar to Lloyd quantizer
- Opt. VQ with variable-length codes: Similar to EC-Lloyd quantizer

Vector Quantizer Advantages

- Space-filling advantage: Unique to vector quantizers (1.53 dB for $N \to \infty$)
- Shape advantage: Can also be exploited by ECSQ
- Memory advantage: Can also be exploited by other coding techniques

Vector Quantization can achieve Rate-Distortion Bound! – Are we done?

➡ No! – Complexity of vector quanzization is a serious issue!

Summary of Lecture

Vector Quantization (VQ)

- Straightforward extension of scalar quantization to higher dimensions N
- Opt. VQ with fixed-length codes: Similar to Lloyd quantizer
- Opt. VQ with variable-length codes: Similar to EC-Lloyd quantizer

Vector Quantizer Advantages

- Space-filling advantage: Unique to vector quantizers (1.53 dB for $N \to \infty$)
- Shape advantage: Can also be exploited by ECSQ
- Memory advantage: Can also be exploited by other coding techniques

Vector Quantization can achieve Rate-Distortion Bound! – Are we done?

➡ No! – Complexity of vector quanzization is a serious issue!

➡ **Require lossy coding techniques with high rate-distortion efficiency and a complexity suitable for wide range of implementations**

Summary of Lecture

Vector Quantization (VQ)

- Straightforward extension of scalar quantization to higher dimensions N
- Opt. VQ with fixed-length codes: Similar to Lloyd quantizer
- Opt. VQ with variable-length codes: Similar to EC-Lloyd quantizer

Vector Quantizer Advantages

- Space-filling advantage: Unique to vector quantizers (1.53 dB for $N \to \infty$)
- Shape advantage: Can also be exploited by ECSQ
- Memory advantage: Can also be exploited by other coding techniques

Vector Quantization can achieve Rate-Distortion Bound! – Are we done?

➡ No! – Complexity of vector quanzization is a serious issue!

➡ **Require lossy coding techniques with high rate-distortion efficiency and a complexity suitable for wide range of implementations**

➡ Particularly important: **Exploitation of dependencies between samples!**

Exercise 1: Space-Filling Gain for 2-dimensional Vector Quantizer

Calculate the gain (in signal-to-noise ratio) of optimal 2-dimensional vector quantization relative to optimal scalar quantization for high rates on the example of a uniform pdf.

Hints:

- In two dimensions, the optimal quantization cells are regular hexagons; the associated reconstruction vectors are located in the centers of the hexagons.
- For high rates, border effects can be neglected. It can be assumed that the signal space for which the pdf is non-zero is completely filled with regular quantization cells.