Algorithms for Optimal Control with Fixed-Rate Feedback

Anatoly Khina

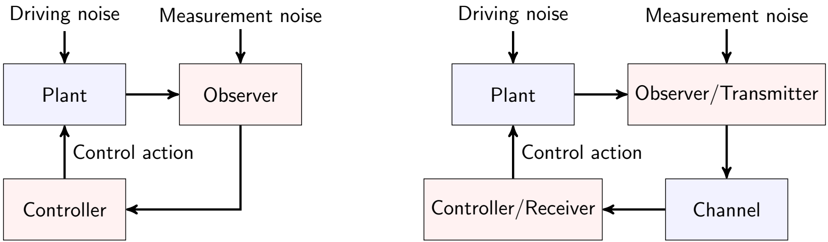
Joint work with Yorie Nakahira, Yu Su, and Babak Hassibi

CDC 2017
Melbourne, VIC, Australia
December 14, 2017

Traditional versus Networked Control

Traditional control

Networked control

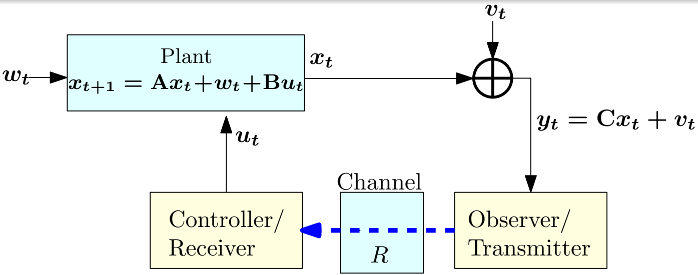


Linear Quadratic Gaussian Control over Gaussian Channels

Linear quadratic Gaussian (LQG) system

$$\mathbf{x}_{t+1} = \mathbf{A}\mathbf{x}_t + \mathbf{B}\mathbf{u}_t + \mathbf{w}_t, \qquad \mathbf{w}_t \sim \text{ i.i.d. } \mathcal{N}(0, \mathbf{W})$$

$$\mathbf{y}_t = \mathbf{C}\mathbf{x}_t + \mathbf{v}_t, \qquad \mathbf{v}_t \sim \text{ i.i.d. } \mathcal{N}(0, \mathbf{V})$$



Noiseless finite-rate channel of rate R

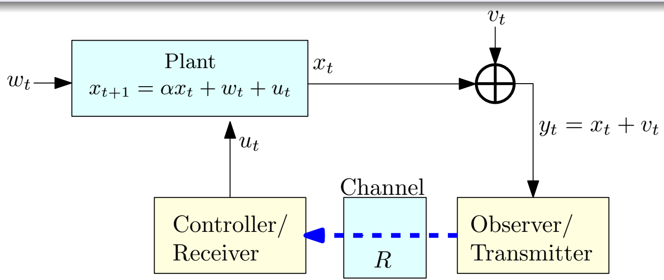
Fixed rate: Exactly R bits are available at every time sample t

Variable rate: R bits are available **on average** at every t

Linear Quadratic Gaussian Control over Gaussian Channels

Scalar Linear quadratic Gaussian (LQG) system

$$x_{t+1} = \alpha x_t + u_t + w_t, \qquad w_t \sim \text{ i.i.d. } \mathcal{N}(0, W), \qquad |\alpha| > 1$$
$$y_t = x_t + v_t, \qquad v_t \sim \text{ i.i.d. } \mathcal{N}(0, V)$$



Noiseless finite-rate channel of rate R

Fixed rate: Exactly R bits are available at every time sample t

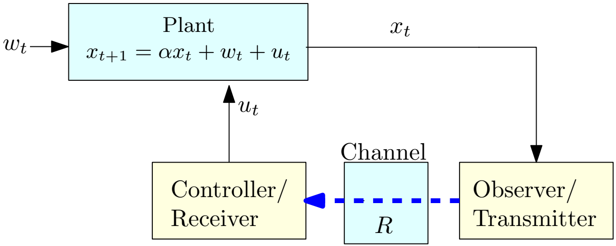
Variable rate: R bits are available **on average** at every t

Linear Quadratic Gaussian Control over Gaussian Channels

Scalar Linear quadratic Gaussian (LQG) system

$$x_{t+1} = \alpha x_t + u_t + w_t, \qquad w_t \sim \text{ i.i.d. } \mathcal{N}(0, W), \qquad |\alpha| > 1$$

$$y_t = x_t + \cancel{v_t}, \qquad \cancel{v_t} \sim \cancel{\text{i.i.d. } \mathcal{N}(0, V)}$$



LQG cost

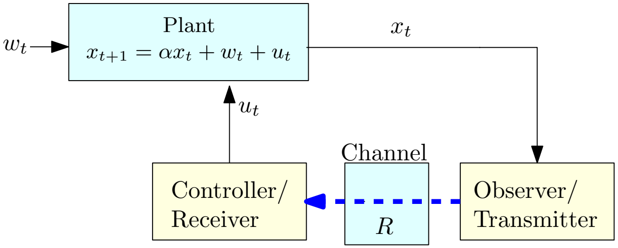
$$\bar{J}_T = \frac{1}{T}\mathbb{E}\left[\sum_{t=1}^{T-1}\left(Q_t x_t^2 + R_t u_t^2\right) + Q_T x_T^2\right]$$

Linear Quadratic Gaussian Control over Gaussian Channels

Scalar Linear quadratic Gaussian (LQG) system

$$x_{t+1} = \alpha x_t + u_t + w_t, \qquad w_t \sim \text{ i.i.d. } \mathcal{N}(0, W), \qquad |\alpha| > 1$$

$$y_t = x_t + \cancel{v_t}, \qquad \cancel{v_t} \sim \cancel{\text{i.i.d. } \mathcal{N}(0, V)}$$



LQG cost: MMSE ($Q_t \equiv 1, R_t \equiv 0$)

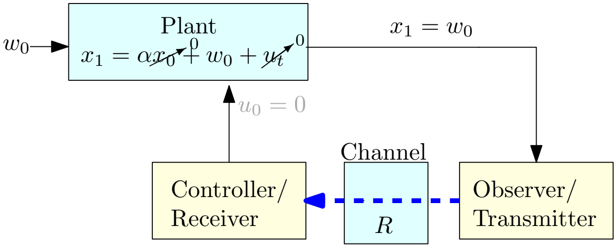
$$\bar{J} = \frac{1}{T}\mathbb{E}\left[\sum_{t=1}^{T} x_t^2\right]$$

Linear Quadratic Gaussian Control over Gaussian Channels

Scalar Linear quadratic Gaussian (LQG) system

$$x_{t+1} = \alpha x_t + u_t + w_t, \qquad w_t \sim \text{ i.i.d. } \mathcal{N}(0, W), \qquad |\alpha| > 1$$

$$y_t = x_t + \cancel{v_t}, \qquad \cancel{v_t} \sim \cancel{\text{ i.i.d. } \mathcal{N}(0, V)}$$



LQG cost: MMSE ($Q_t \equiv 1, R_t \equiv 0$)

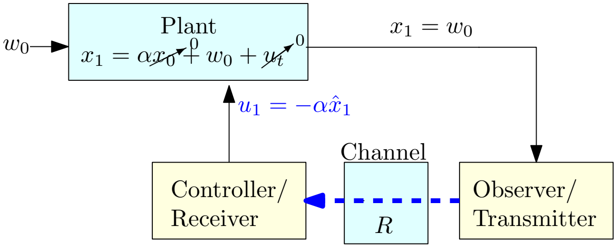
$$\bar{J} = \frac{1}{T} \mathbb{E}\left[\sum_{t=1}^{T} x_t^2\right]$$

Linear Quadratic Gaussian Control over Gaussian Channels

Scalar Linear quadratic Gaussian (LQG) system

$$x_{t+1} = \alpha x_t + u_t + w_t, \qquad w_t \sim \text{ i.i.d. } \mathcal{N}(0, W), \qquad |\alpha| > 1$$

$$y_t = x_t + \cancel{v_t}, \qquad \cancel{v_t} \sim \cancel{\text{i.i.d. } \mathcal{N}(0, V)}$$



LQG cost: MMSE ($Q_t \equiv 1, R_t \equiv 0$)

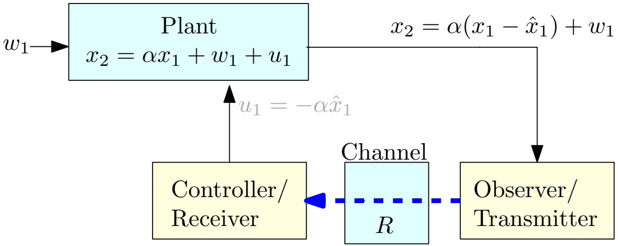
$$\bar{J} = \frac{1}{T}\mathbb{E}\left[\sum_{t=1}^{T} x_t^2\right]$$

Linear Quadratic Gaussian Control over Gaussian Channels

Scalar Linear quadratic Gaussian (LQG) system

$$x_{t+1} = \alpha x_t + u_t + w_t, \qquad w_t \sim \text{ i.i.d. } \mathcal{N}(0, W), \qquad |\alpha| > 1$$

$$y_t = x_t + \cancel{v_t}, \qquad \cancel{v_t} \sim \cancel{\text{i.i.d. } \mathcal{N}(0, V)}$$



LQG cost: MMSE ($Q_t \equiv 1, R_t \equiv 0$)

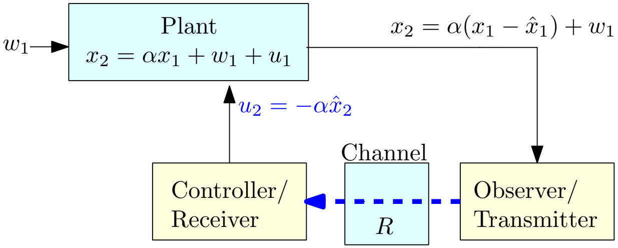
$$\bar{J} = \frac{1}{T} \mathbb{E}\left[\sum_{t=1}^{T} x_t^2\right]$$

Linear Quadratic Gaussian Control over Gaussian Channels

Scalar Linear quadratic Gaussian (LQG) system

$$x_{t+1} = \alpha x_t + u_t + w_t, \qquad w_t \sim \text{ i.i.d. } \mathcal{N}(0, W), \qquad |\alpha| > 1$$

$$y_t = x_t + \cancel{v_t}, \qquad \cancel{v_t} \sim \cancel{\text{i.i.d. } \mathcal{N}(0, V)}$$



LQG cost: MMSE ($Q_t \equiv 1, R_t \equiv 0$)

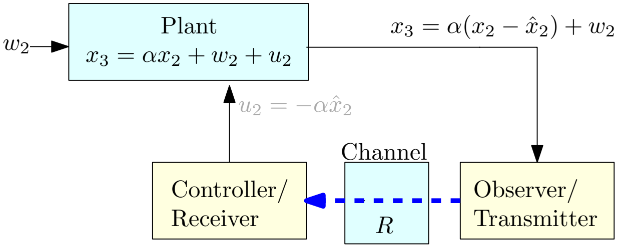
$$\bar{J} = \frac{1}{T}\mathbb{E}\left[\sum_{t=1}^{T} x_t^2\right]$$

Linear Quadratic Gaussian Control over Gaussian Channels

Scalar Linear quadratic Gaussian (LQG) system

$$x_{t+1} = \alpha x_t + u_t + w_t, \qquad w_t \sim \text{ i.i.d. } \mathcal{N}(0, W), \qquad |\alpha| > 1$$

$$y_t = x_t + \sout{v_t}, \qquad \sout{v_t \sim \text{ i.i.d. } \mathcal{N}(0, V)}$$



LQG cost: MMSE ($Q_t \equiv 1, R_t \equiv 0$)

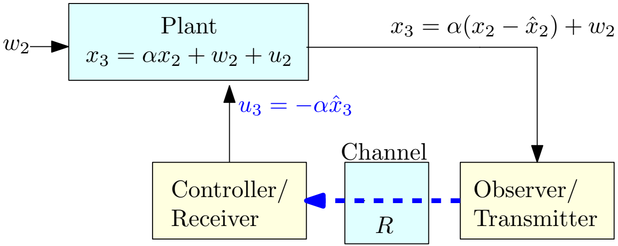
$$\bar{J} = \frac{1}{T}\mathbb{E}\left[\sum_{t=1}^{T} x_t^2\right]$$

Linear Quadratic Gaussian Control over Gaussian Channels

Scalar Linear quadratic Gaussian (LQG) system

$$x_{t+1} = \alpha x_t + u_t + w_t, \qquad w_t \sim \text{i.i.d. } \mathcal{N}(0, W), \qquad |\alpha| > 1$$

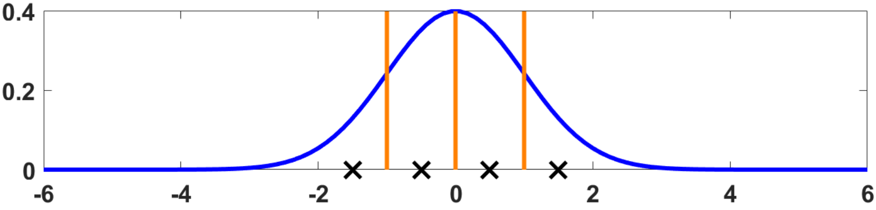
$$y_t = x_t + \cancel{v_t}, \qquad \cancel{v_t \sim \text{i.i.d. } \mathcal{N}(0, V)}$$



LQG cost: MMSE ($Q_t \equiv 1, R_t \equiv 0$)

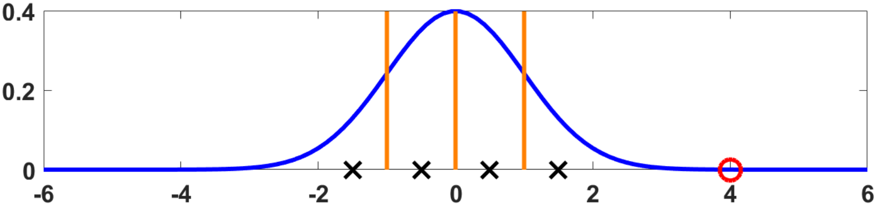
$$\bar{J} = \frac{1}{T}\mathbb{E}\left[\sum_{t=1}^{T} x_t^2\right]$$

Adaptive Fixed-Rate Quantizer



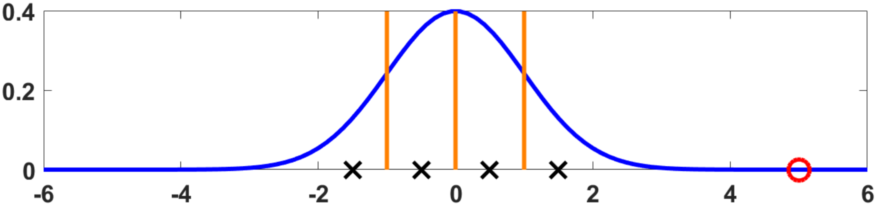
- Use an adjusted quantizer to the input p.d.f.

Adaptive Fixed-Rate Quantizer



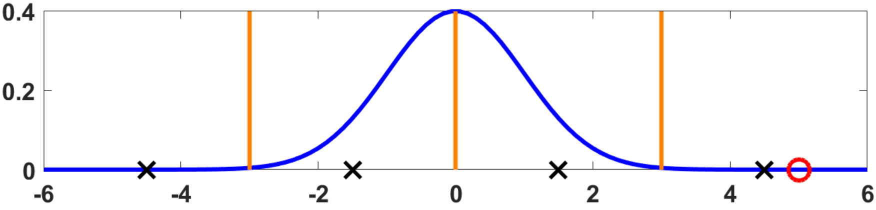
- Use an adjusted quantizer to the input p.d.f.
- At some point a (rare) event will happen
- Input value outside effective quantization interval

Adaptive Fixed-Rate Quantizer



- Use an adjusted quantizer to the input p.d.f.
- At some point a (rare) event will happen
- Input value outside effective quantization interval
- Next time instant: Input will be even larger!
- **Avalanche effect**

Adaptive Fixed-Rate Quantizer



- Use an adjusted quantizer to the input p.d.f.
- At some point a (rare) event will happen
- Input value outside effective quantization interval
- Next time instant: Input will be even larger!
- **Avalanche effect**
- To avoid this $\Rightarrow$ Quantizer needs to be **adaptive**

Adaptive Optimal Fixed-Rate Quantizer?

- Adaptive uniform quantizer [Yüksel AC'10]
  - Based on Jayant's adaptive quantizer [Jayant BLTJ'73]
  - Similar idea in [Brockett-Liberzon AC'00]: "Zooming in/out"
- Adaptive exponential quantizer [Minero et al. AC'09]
- Both results prove condition on stabilizability: $R > \log \alpha$
- But no cost optimality claims...
- Notable contributions: [Borkar-Mitter '97][Matveev-Savkin '04] [Tatikonda-Sahai-Mitter AC'04] [Tsumura-Maciejowski CDC'03] [Linder-Yüksel IT'14] [Yüksel AC'14] ...

How to optimize cost?

Adaptive Optimal Fixed-Rate Quantizer?

- Adaptive uniform quantizer [Yüksel AC'10]
  - Based on Jayant's adaptive quantizer [Jayant BLTJ'73]
  - Similar idea in [Brockett-Liberzon AC'00]: "Zooming in/out"
- Adaptive exponential quantizer [Minero et al. AC'09]
- Both results prove condition on stabilizability: $R > \log \alpha$
- But no cost optimality claims...
- Notable contributions: [Borkar-Mitter '97][Matveev-Savkin '04] [Tatikonda-Sahai-Mitter AC'04] [Tsumura-Maciejowski CDC'03] [Linder-Yüksel IT'14] [Yüksel AC'14] ...

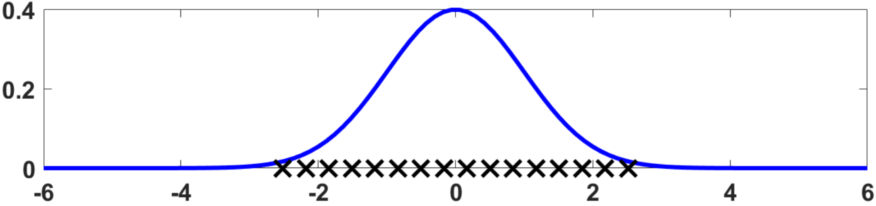
How to optimize cost?

Idea [Bao-Skoglund-Johansson AC'11][Nakahira CDC'16]

- Use the Lloyd–Max algorithm
- **Is it optimal?**

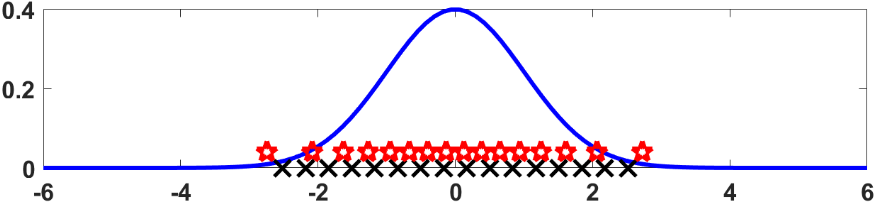
Optimal Quantizer for One Sample

- Let $x \sim \mathcal{N}(0,1)$
- R bits $\Rightarrow 2^R$ quantization points
- Uniform quantizer is suboptimal
- How to construct an optimal quantizer?



Optimal Quantizer for One Sample

- Let $x \sim \mathcal{N}(0,1)$
- R bits $\Rightarrow 2^R$ quantization points
- Uniform quantizer is suboptimal
- How to construct an optimal quantizer?



- In general a hard (NP-hard) problem
- Necessary conditions by Lloyd ['57, IT'82] and Max [IT '60]
  - Also known in machine learning as "k-means" clustering

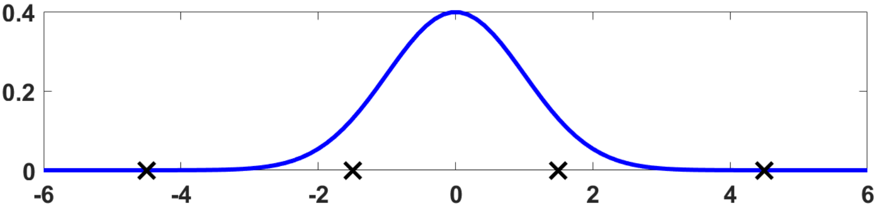
Lloyd-Max Algorithm

Nearest Neighbor: Given reconstruction points, find optimal cells

$$\text{Cell } i = \left\{x \middle| (x - \hat{x}_i)^2 < (x - \hat{x}_j)^2, \ \forall j \neq i\right\}$$

Centroid: Given quant. cells, find optimal reconstruction points

$$\hat{x}_i = \mathbb{E}\left[x \middle| x \in \text{Cell } i\right]$$



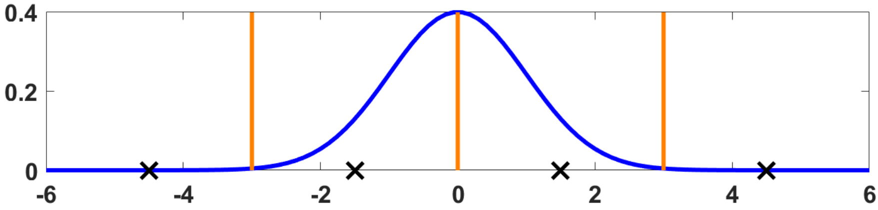
Lloyd-Max Algorithm

Nearest Neighbor: Given reconstruction points, find optimal cells

$$\text{Cell } i = \left\{x \middle| (x - \hat{x}_i)^2 < (x - \hat{x}_j)^2, \ \forall j \neq i \right\}$$

Centroid: Given quant. cells, find optimal reconstruction points

$$\hat{x}_i = \mathbb{E}\left[x \middle| x \in \text{Cell } i\right]$$



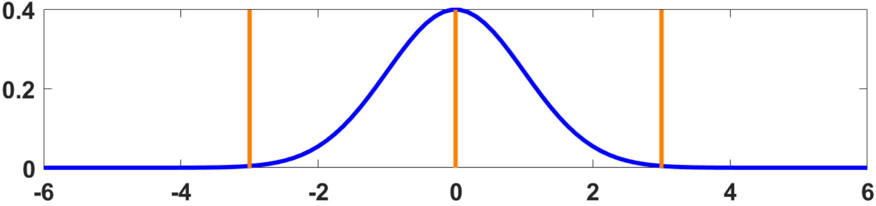
Lloyd-Max Algorithm

Nearest Neighbor: Given reconstruction points, find optimal cells

$$\text{Cell } i = \left\{ x \middle| (x - \hat{x}_i)^2 < (x - \hat{x}_j)^2, \ \forall j \neq i \right\}$$

Centroid: Given quant. cells, find optimal reconstruction points

$$\hat{x}_i = \mathbb{E}\left[x | x \in \text{Cell } i \right]$$



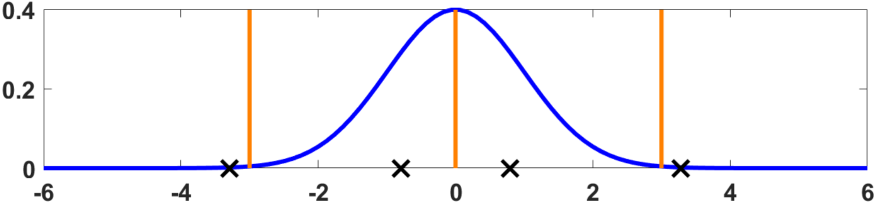
Lloyd-Max Algorithm

Nearest Neighbor: Given reconstruction points, find optimal cells

$$\text{Cell } i = \left\{x \middle| (x - \hat{x}_i)^2 < (x - \hat{x}_j)^2, \ \forall j \neq i\right\}$$

Centroid: Given quant. cells, find optimal reconstruction points

$$\hat{x}_i = \mathbb{E}\left[x \middle| x \in \text{Cell } i\right]$$



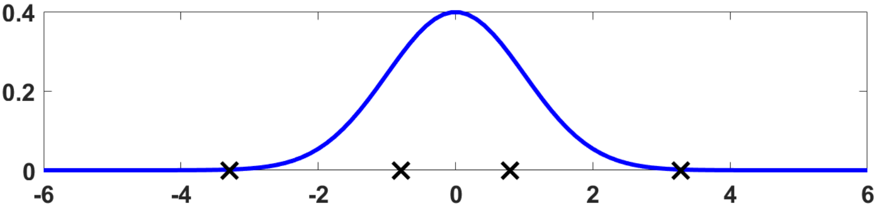
Lloyd-Max Algorithm

Nearest Neighbor: Given reconstruction points, find optimal cells

$$\text{Cell } i = \left\{x \middle| (x - \hat{x}_i)^2 < (x - \hat{x}_j)^2, \ \forall j \neq i\right\}$$

Centroid: Given quant. cells, find optimal reconstruction points

$$\hat{x}_i = \mathbb{E}\left[x | x \in \text{Cell } i\right]$$



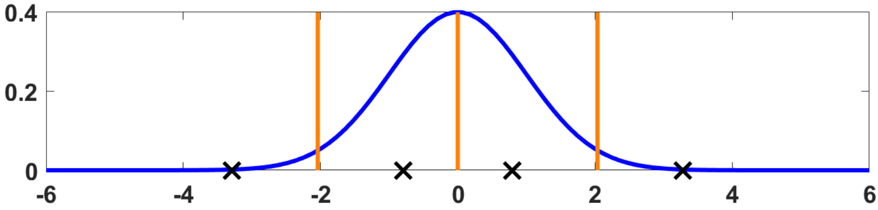
Lloyd-Max Algorithm

Nearest Neighbor: Given reconstruction points, find optimal cells

$$\text{Cell } i = \left\{ x \middle| (x - \hat{x}_i)^2 < (x - \hat{x}_j)^2, \ \forall j \neq i \right\}$$

Centroid: Given quant. cells, find optimal reconstruction points

$$\hat{x}_i = \mathbb{E}\left[x \middle| x \in \text{Cell } i \right]$$



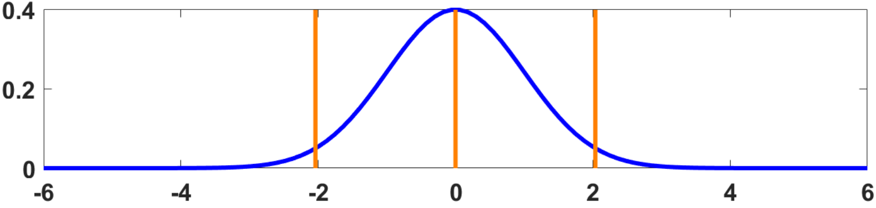
Lloyd-Max Algorithm

Nearest Neighbor: Given reconstruction points, find optimal cells

$$\text{Cell } i = \left\{ x \middle| (x - \hat{x}_i)^2 < (x - \hat{x}_j)^2, \ \forall j \neq i \right\}$$

Centroid: Given quant. cells, find optimal reconstruction points

$$\hat{x}_i = \mathbb{E}\left[x \middle| x \in \text{Cell } i \right]$$



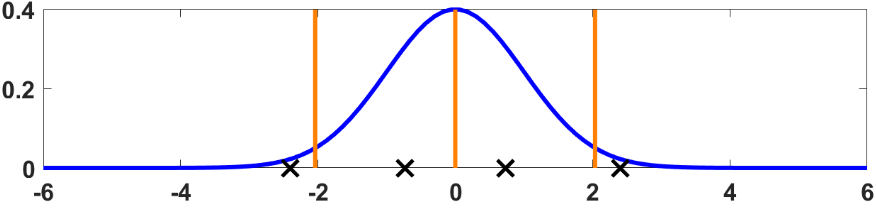
Lloyd-Max Algorithm

Nearest Neighbor: Given reconstruction points, find optimal cells

$$\text{Cell } i = \left\{x \middle| (x - \hat{x}_i)^2 < (x - \hat{x}_j)^2, \ \forall j \neq i\right\}$$

Centroid: Given quant. cells, find optimal reconstruction points

$$\hat{x}_i = \mathbb{E}\left[x \middle| x \in \text{Cell } i\right]$$



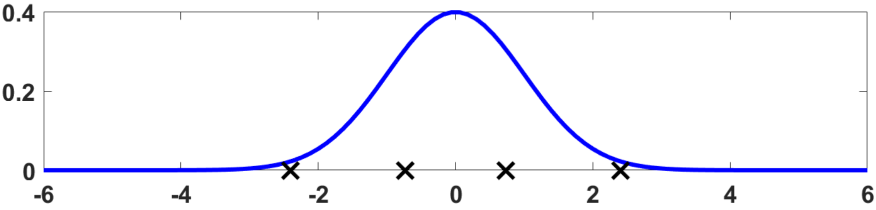
Lloyd-Max Algorithm

Nearest Neighbor: Given reconstruction points, find optimal cells

$$\text{Cell } i = \left\{ x \middle| (x - \hat{x}_i)^2 < (x - \hat{x}_j)^2, \ \forall j \neq i \right\}$$

Centroid: Given quant. cells, find optimal reconstruction points

$$\hat{x}_i = \mathbb{E}\left[x | x \in \text{Cell } i \right]$$



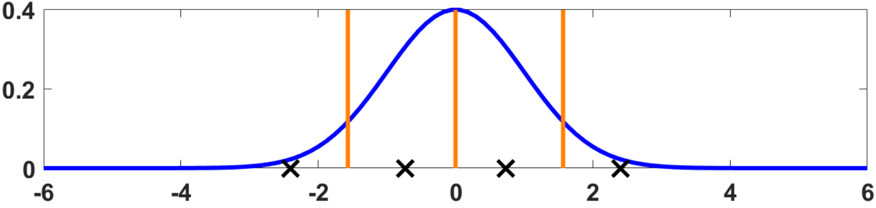
Lloyd-Max Algorithm

Nearest Neighbor: Given reconstruction points, find optimal cells

$$\text{Cell } i = \left\{x \middle| (x - \hat{x}_i)^2 < (x - \hat{x}_j)^2, \ \forall j \neq i\right\}$$

Centroid: Given quant. cells, find optimal reconstruction points

$$\hat{x}_i = \mathbb{E}\left[x | x \in \text{Cell } i\right]$$



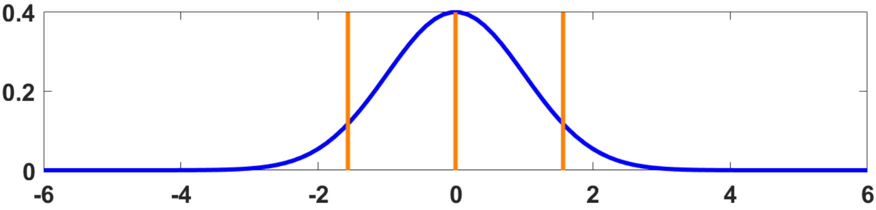
Lloyd-Max Algorithm

Nearest Neighbor: Given reconstruction points, find optimal cells

$$\text{Cell } i = \left\{ x \middle| (x - \hat{x}_i)^2 < (x - \hat{x}_j)^2, \ \forall j \neq i \right\}$$

Centroid: Given quant. cells, find optimal reconstruction points

$$\hat{x}_i = \mathbb{E}\left[x | x \in \text{Cell } i \right]$$



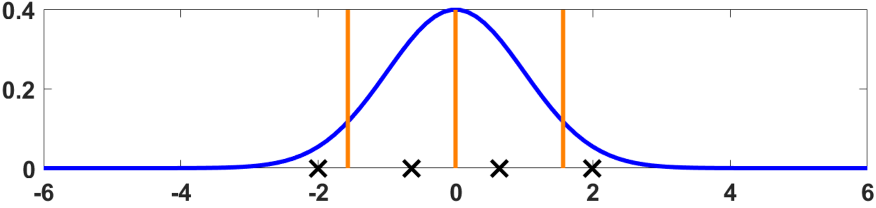
Lloyd-Max Algorithm

Nearest Neighbor: Given reconstruction points, find optimal cells

$$\text{Cell } i = \left\{ x \middle| (x - \hat{x}_i)^2 < (x - \hat{x}_j)^2, \ \forall j \neq i \right\}$$

Centroid: Given quant. cells, find optimal reconstruction points

$$\hat{x}_i = \mathbb{E}\left[x \middle| x \in \text{Cell } i \right]$$



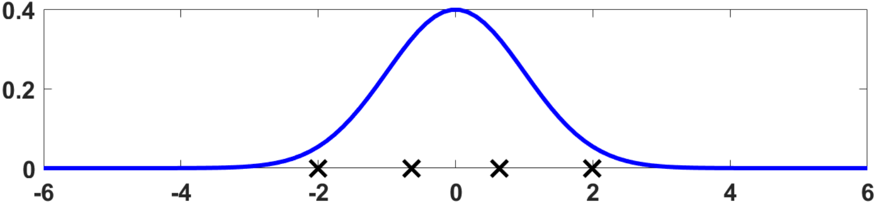
Lloyd-Max Algorithm

Nearest Neighbor: Given reconstruction points, find optimal cells

$$\text{Cell } i = \left\{ x \middle| (x - \hat{x}_i)^2 < (x - \hat{x}_j)^2 , \ \forall j \neq i \right\}$$

Centroid: Given quant. cells, find optimal reconstruction points

$$\hat{x}_i = \mathbb{E}\left[x | x \in \text{Cell } i \right]$$



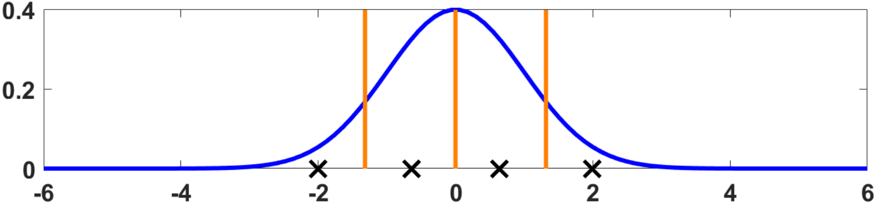
Lloyd-Max Algorithm

Nearest Neighbor: Given reconstruction points, find optimal cells

$$\text{Cell } i = \left\{ x \middle| (x - \hat{x}_i)^2 < (x - \hat{x}_j)^2, \ \forall j \neq i \right\}$$

Centroid: Given quant. cells, find optimal reconstruction points

$$\hat{x}_i = \mathbb{E}\left[x | x \in \text{Cell } i \right]$$



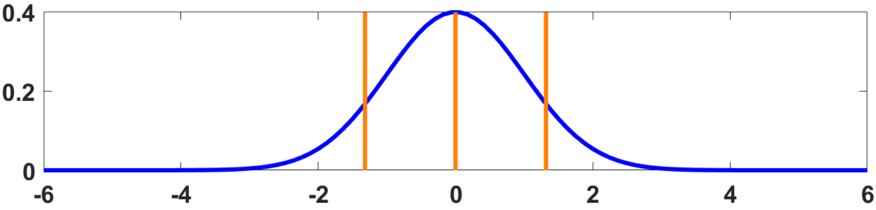
Lloyd-Max Algorithm

Nearest Neighbor: Given reconstruction points, find optimal cells

$$\text{Cell } i = \left\{x \middle| (x - \hat{x}_i)^2 < (x - \hat{x}_j)^2,\ \forall j \neq i\right\}$$

Centroid: Given quant. cells, find optimal reconstruction points

$$\hat{x}_i = \mathbb{E}\left[x \middle| x \in \text{Cell } i\right]$$



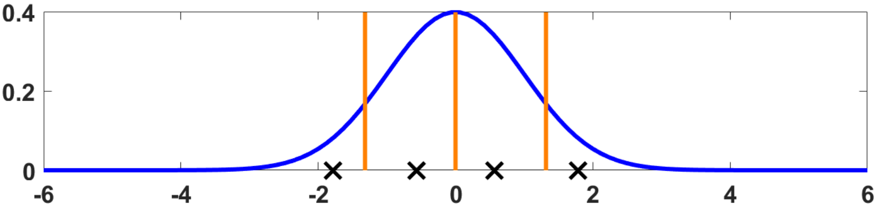
Lloyd-Max Algorithm

Nearest Neighbor: Given reconstruction points, find optimal cells

$$\text{Cell } i = \left\{ x \middle| (x - \hat{x}_i)^2 < (x - \hat{x}_j)^2, \ \forall j \neq i \right\}$$

Centroid: Given quant. cells, find optimal reconstruction points

$$\hat{x}_i = \mathbb{E}\left[x \middle| x \in \text{Cell } i \right]$$



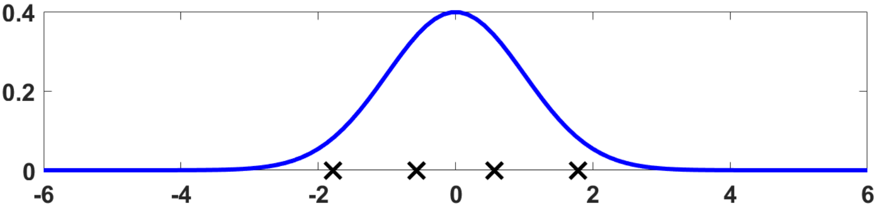
Lloyd-Max Algorithm

Nearest Neighbor: Given reconstruction points, find optimal cells

$$\text{Cell } i = \left\{ x \middle| (x - \hat{x}_i)^2 < (x - \hat{x}_j)^2, \; \forall j \neq i \right\}$$

Centroid: Given quant. cells, find optimal reconstruction points

$$\hat{x}_i = \mathbb{E}\left[x | x \in \text{Cell } i \right]$$



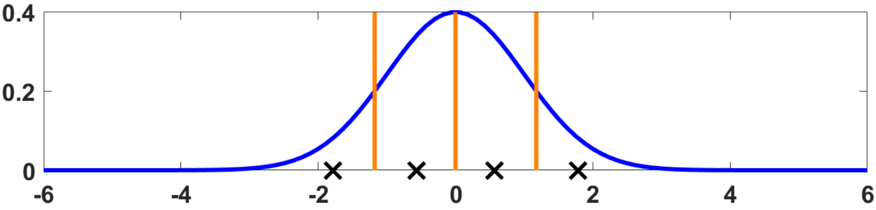
Lloyd-Max Algorithm

Nearest Neighbor: Given reconstruction points, find optimal cells

$$\text{Cell } i = \left\{x \middle| (x - \hat{x}_i)^2 < (x - \hat{x}_j)^2,\ \forall j \neq i\right\}$$

Centroid: Given quant. cells, find optimal reconstruction points

$$\hat{x}_i = \mathbb{E}\left[x \middle| x \in \text{Cell } i\right]$$



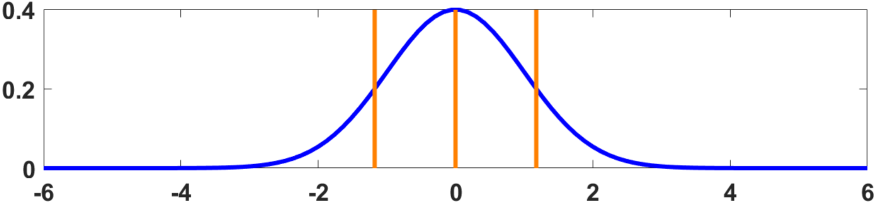
Lloyd-Max Algorithm

Nearest Neighbor: Given reconstruction points, find optimal cells

$$\text{Cell } i = \left\{ x \middle| (x - \hat{x}_i)^2 < (x - \hat{x}_j)^2, \ \forall j \neq i \right\}$$

Centroid: Given quant. cells, find optimal reconstruction points

$$\hat{x}_i = \mathbb{E}\left[x | x \in \text{Cell } i \right]$$



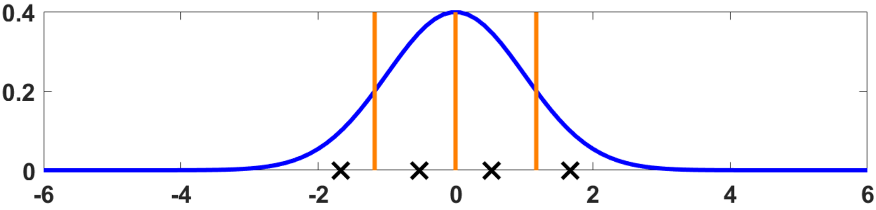
Lloyd-Max Algorithm

Nearest Neighbor: Given reconstruction points, find optimal cells

$$\text{Cell } i = \left\{ x \middle| (x - \hat{x}_i)^2 < (x - \hat{x}_j)^2, \ \forall j \neq i \right\}$$

Centroid: Given quant. cells, find optimal reconstruction points

$$\hat{x}_i = \mathbb{E}\left[x | x \in \text{Cell } i \right]$$



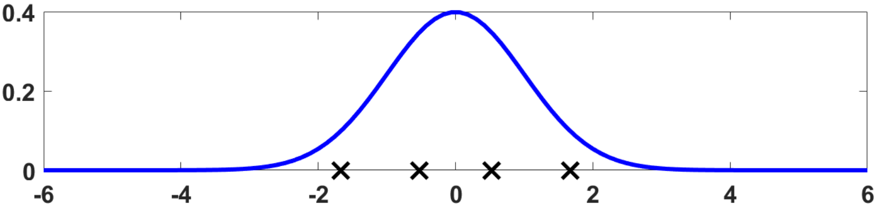
Lloyd-Max Algorithm

Nearest Neighbor: Given reconstruction points, find optimal cells

$$\text{Cell } i = \left\{ x \middle| (x - \hat{x}_i)^2 < (x - \hat{x}_j)^2, \ \forall j \neq i \right\}$$

Centroid: Given quant. cells, find optimal reconstruction points

$$\hat{x}_i = \mathbb{E}\left[x \middle| x \in \text{Cell } i \right]$$



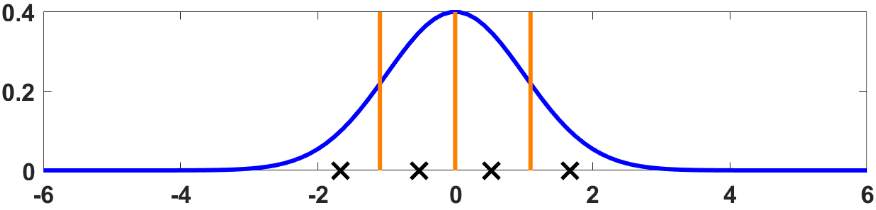
Lloyd-Max Algorithm

Nearest Neighbor: Given reconstruction points, find optimal cells

$$\text{Cell } i = \left\{ x \middle| (x - \hat{x}_i)^2 < (x - \hat{x}_j)^2, \ \forall j \neq i \right\}$$

Centroid: Given quant. cells, find optimal reconstruction points

$$\hat{x}_i = \mathbb{E}\left[x | x \in \text{Cell } i \right]$$



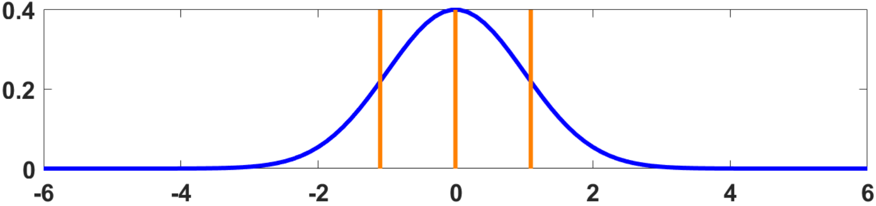
Lloyd-Max Algorithm

Nearest Neighbor: Given reconstruction points, find optimal cells

$$\text{Cell } i = \left\{ x \middle| (x - \hat{x}_i)^2 < (x - \hat{x}_j)^2, \ \forall j \neq i \right\}$$

Centroid: Given quant. cells, find optimal reconstruction points

$$\hat{x}_i = \mathbb{E}\left[x | x \in \text{Cell } i \right]$$



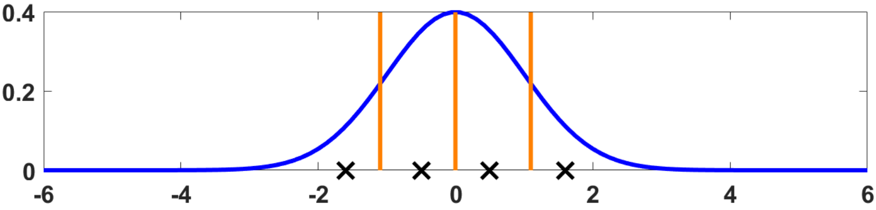
Lloyd-Max Algorithm

Nearest Neighbor: Given reconstruction points, find optimal cells

$$\text{Cell } i = \left\{ x \middle| (x - \hat{x}_i)^2 < (x - \hat{x}_j)^2, \ \forall j \neq i \right\}$$

Centroid: Given quant. cells, find optimal reconstruction points

$$\hat{x}_i = \mathbb{E}\left[x \middle| x \in \text{Cell } i \right]$$



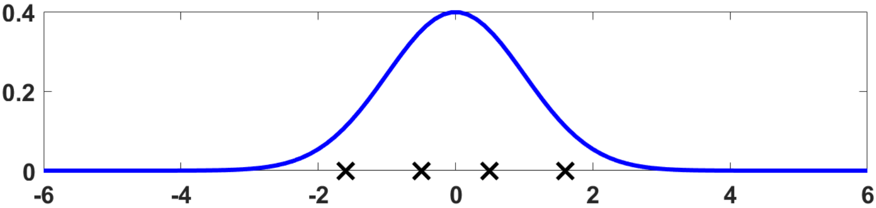
Lloyd-Max Algorithm

Nearest Neighbor: Given reconstruction points, find optimal cells

$$\text{Cell } i = \left\{ x \middle| (x - \hat{x}_i)^2 < (x - \hat{x}_j)^2, \ \forall j \neq i \right\}$$

Centroid: Given quant. cells, find optimal reconstruction points

$$\hat{x}_i = \mathbb{E}\left[x | x \in \text{Cell } i \right]$$



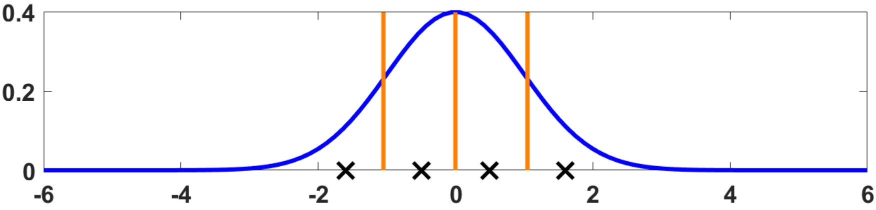
Lloyd-Max Algorithm

Nearest Neighbor: Given reconstruction points, find optimal cells

$$\text{Cell } i = \left\{ x \middle| (x - \hat{x}_i)^2 < (x - \hat{x}_j)^2, \ \forall j \neq i \right\}$$

Centroid: Given quant. cells, find optimal reconstruction points

$$\hat{x}_i = \mathbb{E}\left[x \middle| x \in \text{Cell } i \right]$$



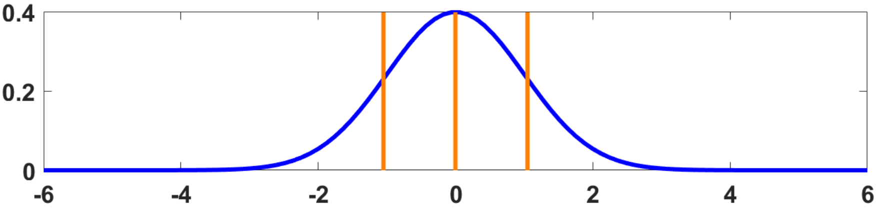
Lloyd-Max Algorithm

Nearest Neighbor: Given reconstruction points, find optimal cells

$$\text{Cell } i = \left\{x \middle| (x - \hat{x}_i)^2 < (x - \hat{x}_j)^2, \ \forall j \neq i \right\}$$

Centroid: Given quant. cells, find optimal reconstruction points

$$\hat{x}_i = \mathbb{E}\left[x \middle| x \in \text{Cell } i\right]$$



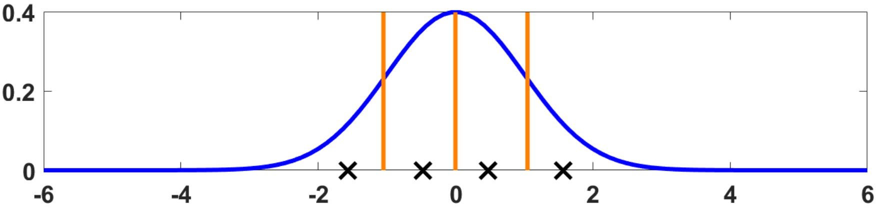
Lloyd-Max Algorithm

Nearest Neighbor: Given reconstruction points, find optimal cells

$$\text{Cell } i = \left\{ x \middle| (x - \hat{x}_i)^2 < (x - \hat{x}_j)^2, \ \forall j \neq i \right\}$$

Centroid: Given quant. cells, find optimal reconstruction points

$$\hat{x}_i = \mathbb{E}\left[x | x \in \text{Cell } i \right]$$



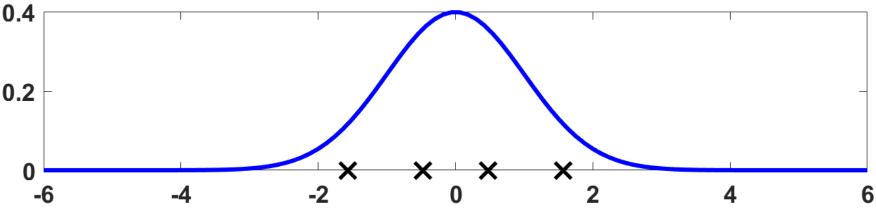
Lloyd-Max Algorithm

Nearest Neighbor: Given reconstruction points, find optimal cells

$$\text{Cell } i = \left\{x \middle| (x - \hat{x}_i)^2 < (x - \hat{x}_j)^2, \ \forall j \neq i\right\}$$

Centroid: Given quant. cells, find optimal reconstruction points

$$\hat{x}_i = \mathbb{E}\left[x \middle| x \in \text{Cell } i\right]$$



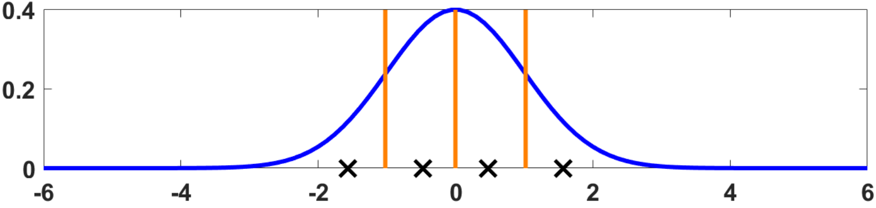
Lloyd-Max Algorithm

Nearest Neighbor: Given reconstruction points, find optimal cells

$$\text{Cell } i = \left\{ x \middle| (x - \hat{x}_i)^2 < (x - \hat{x}_j)^2, \ \forall j \neq i \right\}$$

Centroid: Given quant. cells, find optimal reconstruction points

$$\hat{x}_i = \mathbb{E}\left[x | x \in \text{Cell } i \right]$$



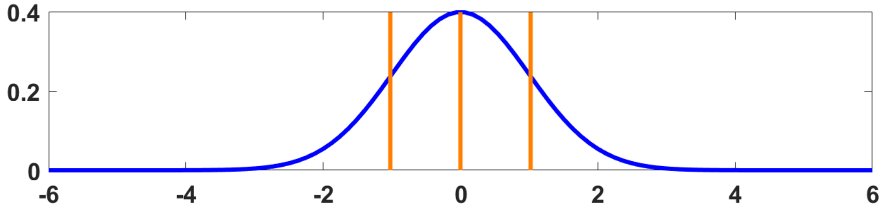
Lloyd-Max Algorithm

Nearest Neighbor: Given reconstruction points, find optimal cells

$$\text{Cell } i = \left\{ x \middle| (x - \hat{x}_i)^2 < (x - \hat{x}_j)^2, \ \forall j \neq i \right\}$$

Centroid: Given quant. cells, find optimal reconstruction points

$$\hat{x}_i = \mathbb{E}\left[x \middle| x \in \text{Cell } i \right]$$



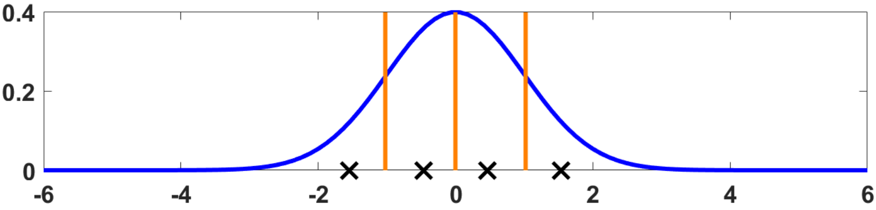
Lloyd-Max Algorithm

Nearest Neighbor: Given reconstruction points, find optimal cells

$$\text{Cell } i = \left\{ x \middle| (x - \hat{x}_i)^2 < (x - \hat{x}_j)^2, \ \forall j \neq i \right\}$$

Centroid: Given quant. cells, find optimal reconstruction points

$$\hat{x}_i = \mathbb{E}\left[x \middle| x \in \text{Cell } i \right]$$



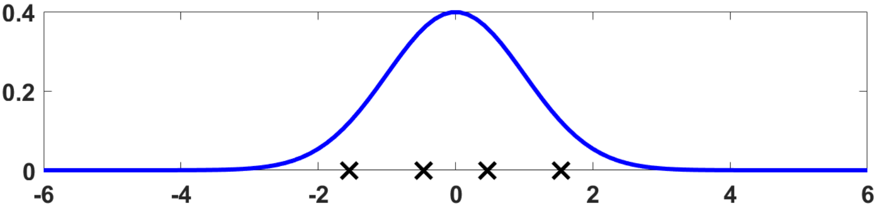
Lloyd-Max Algorithm

Nearest Neighbor: Given reconstruction points, find optimal cells

$$\text{Cell } i = \left\{ x \middle| (x - \hat{x}_i)^2 < (x - \hat{x}_j)^2, \ \forall j \neq i \right\}$$

Centroid: Given quant. cells, find optimal reconstruction points

$$\hat{x}_i = \mathbb{E}\left[x \middle| x \in \text{Cell } i \right]$$



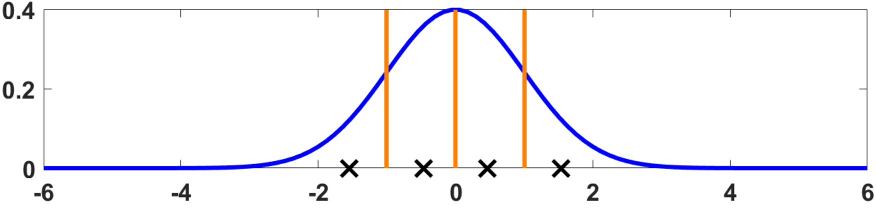
Lloyd-Max Algorithm

Nearest Neighbor: Given reconstruction points, find optimal cells

$$\text{Cell } i = \left\{ x \middle| (x - \hat{x}_i)^2 < (x - \hat{x}_j)^2, \ \forall j \neq i \right\}$$

Centroid: Given quant. cells, find optimal reconstruction points

$$\hat{x}_i = \mathbb{E}\left[x | x \in \text{Cell } i \right]$$



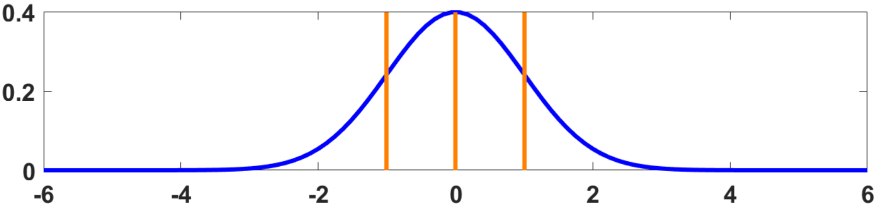
Lloyd-Max Algorithm

Nearest Neighbor: Given reconstruction points, find optimal cells

$$\text{Cell } i = \left\{ x \middle| (x - \hat{x}_i)^2 < (x - \hat{x}_j)^2, \ \forall j \neq i \right\}$$

Centroid: Given quant. cells, find optimal reconstruction points

$$\hat{x}_i = \mathbb{E}\left[x | x \in \text{Cell } i \right]$$



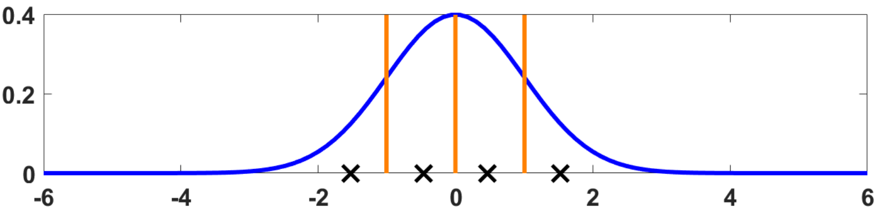
Lloyd-Max Algorithm

Nearest Neighbor: Given reconstruction points, find optimal cells

$$\text{Cell } i = \left\{ x \middle| (x - \hat{x}_i)^2 < (x - \hat{x}_j)^2, \ \forall j \neq i \right\}$$

Centroid: Given quant. cells, find optimal reconstruction points

$$\hat{x}_i = \mathbb{E}\left[x \middle| x \in \text{Cell } i \right]$$



Lloyd-Max Algorithm

Nearest Neighbor: Given reconstruction points, find optimal cells

$$\text{Cell } i = \left\{ x \middle| (x - \hat{x}_i)^2 < (x - \hat{x}_j)^2 ,\ \forall j \neq i \right\}$$

Centroid: Given quant. cells, find optimal reconstruction points

$$\hat{x}_i = \mathbb{E}\left[x | x \in \text{Cell } i \right]$$

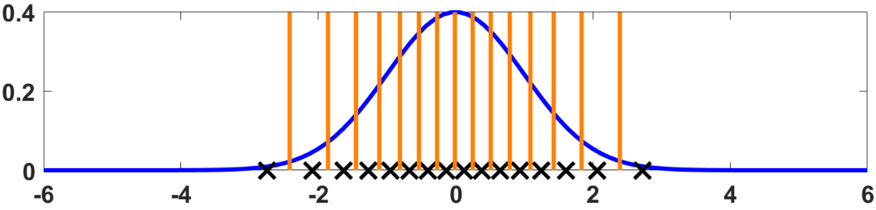
Lloyd-Max Algorithm

Nearest Neighbor: Given reconstruction points, find optimal cells

$$\text{Cell } i = \left\{ x \middle| (x - \hat{x}_i)^2 < (x - \hat{x}_j)^2, \ \forall j \neq i \right\}$$

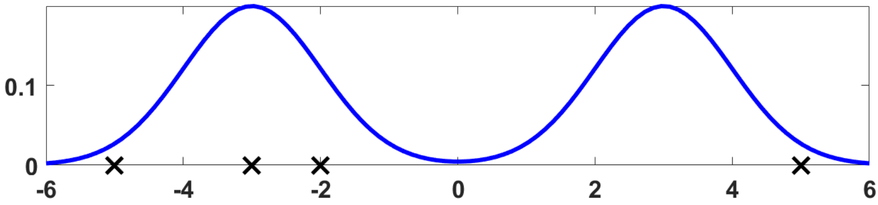
Centroid: Given quant. cells, find optimal reconstruction points

$$\hat{x}_i = \mathbb{E}\left[x | x \in \text{Cell } i \right]$$



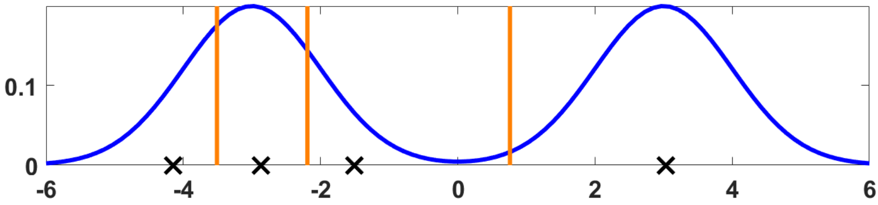
Lloyd-Max Algorithm

- Optimal quantizer necessarily satisfies *Centroid* and *NN*
- But... They are not sufficient in general! ☹
- Lloyd-Max algorithm might converge to a local optimum...



Lloyd-Max Algorithm

- Optimal quantizer necessarily satisfies *Centroid* and *NN*
- But... They are not sufficient in general! ☹
- Lloyd-Max algorithm might converge to a local optimum...



Lloyd-Max Algorithm

When does Lloyd-Max converge to global optimum?
[Fleischer '64][Trushkin IT'82][Kieffer-Jahns-Obuljen IT'88]

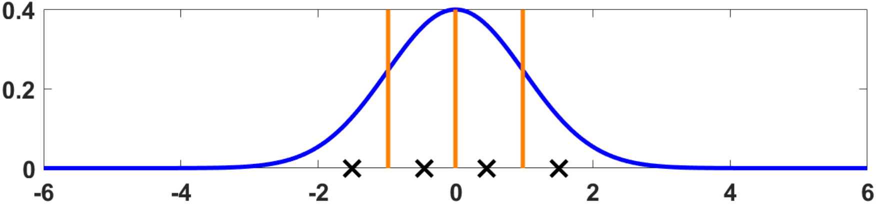
- Conditions for existence of only one local optimum $\Rightarrow$ **Global**
- **Log-concave** distributions satisfy these conditions
- Important special case: **Gaussian distribution** ☺

- One stage of LQG with finite-rate noiseless channel ✓

What about more stages?

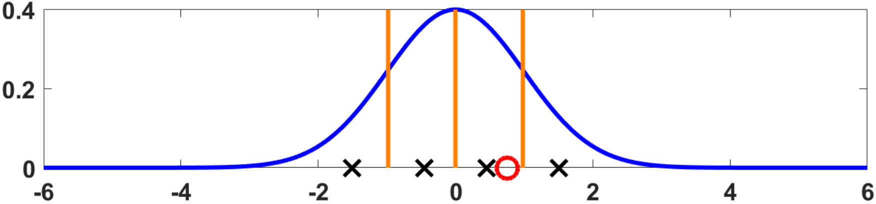
Multi-Stage Control with Finite-Rate Feedback

- First input $x_1 = w_0$ is Gaussian $\Rightarrow$ Log-concave pdf
- Lloyd-Max quantizer is optimal



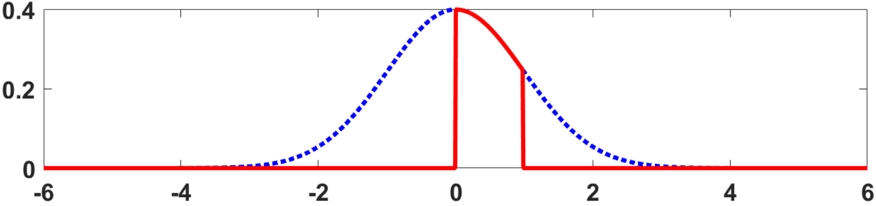
Multi-Stage Control with Finite-Rate Feedback

- First input x_1 arrives
- Chooses cell: cell i
- Chooses reconstruction point: $\hat{x}_i$



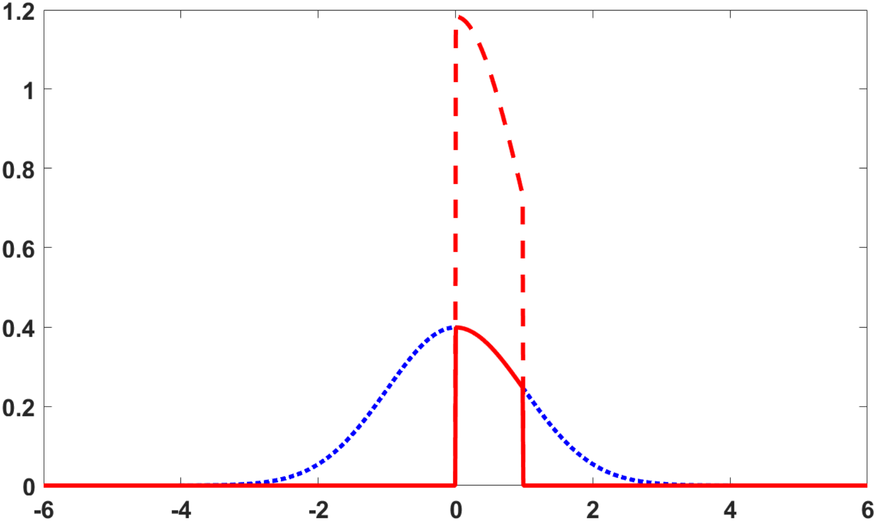
Multi-Stage Control with Finite-Rate Feedback

- $\underset{p(x_1 | x_1 \in \text{cell } i)}{\text{pdf given hit cell } i} = \underset{p(x_1)}{\text{truncated original pdf}}$



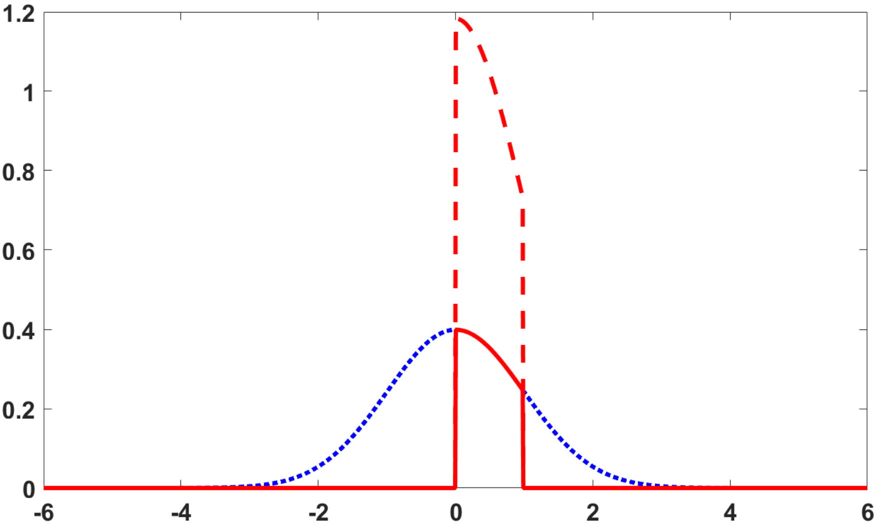
Multi-Stage Control with Finite-Rate Feedback

- Up to scaling...



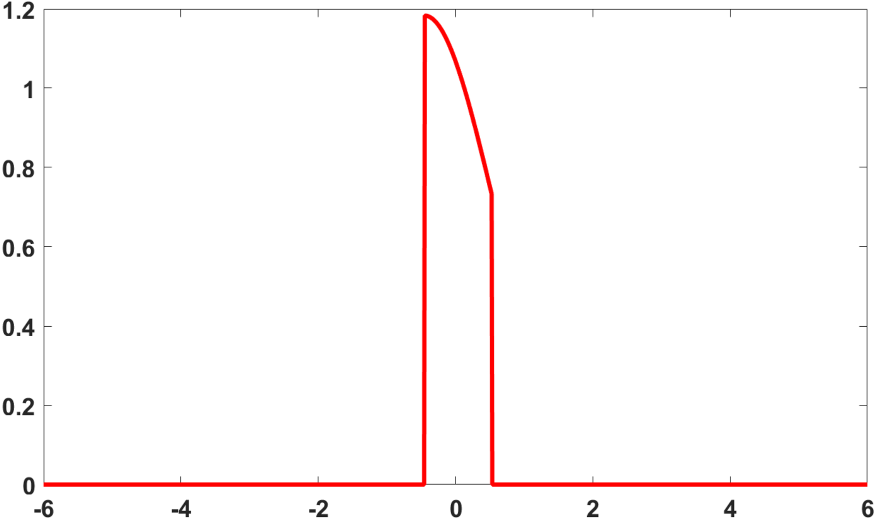
Multi-Stage Control with Finite-Rate Feedback

- Truncated log-concave pdf is **log-concave**!



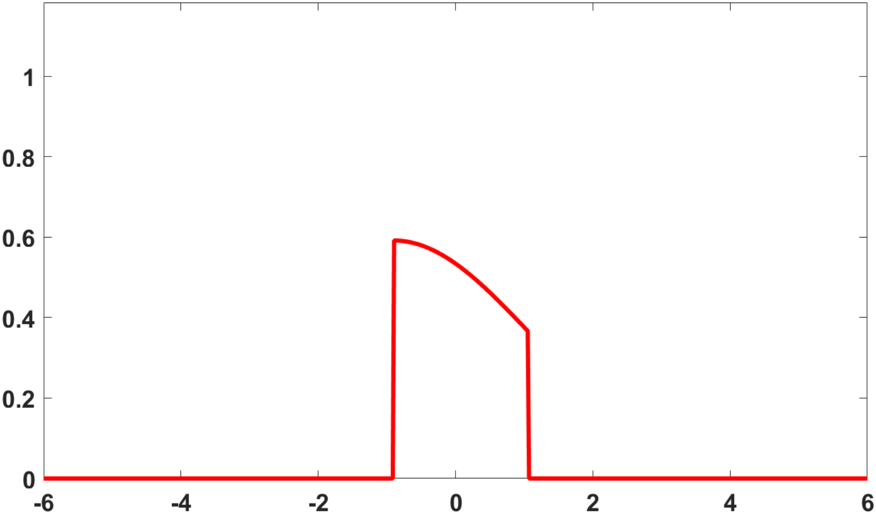
Multi-Stage Control with Finite-Rate Feedback

- pdf of quantization noise $p(x_1 - \hat{x}_1 | x_1 \in \text{ cell } i)$



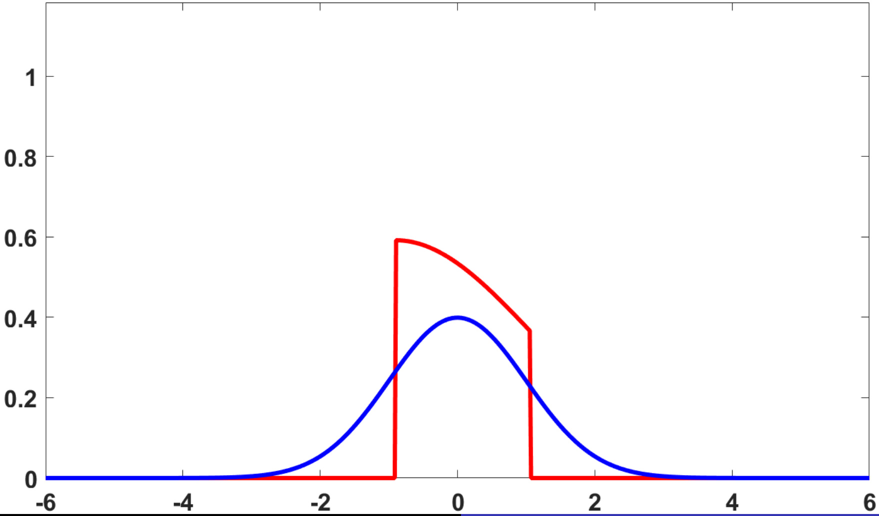
Multi-Stage Control with Finite-Rate Feedback

- Quantization noise inflated by α: $\alpha(x_1 - \hat{x}_1)$



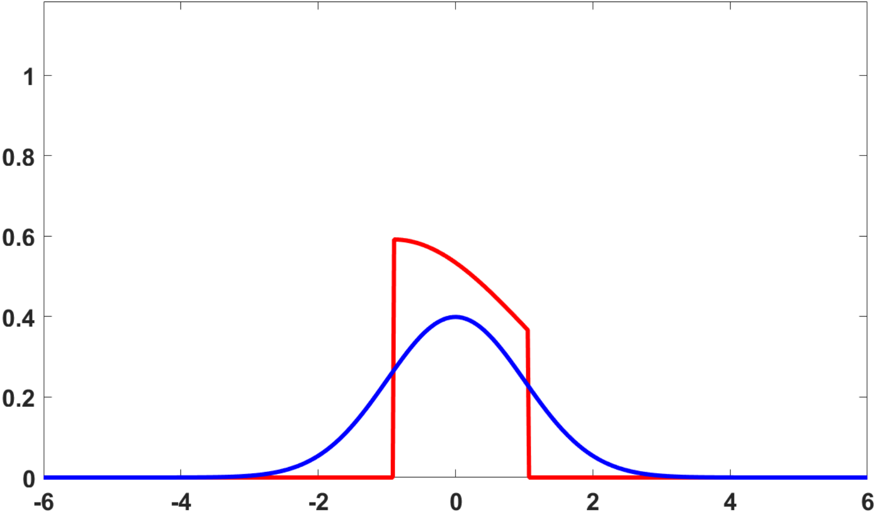
Multi-Stage Control with Finite-Rate Feedback

- New w_t added: $\alpha(x_1 - \hat{x}_1) + w_1 \Rightarrow$ Convolution of pdfs



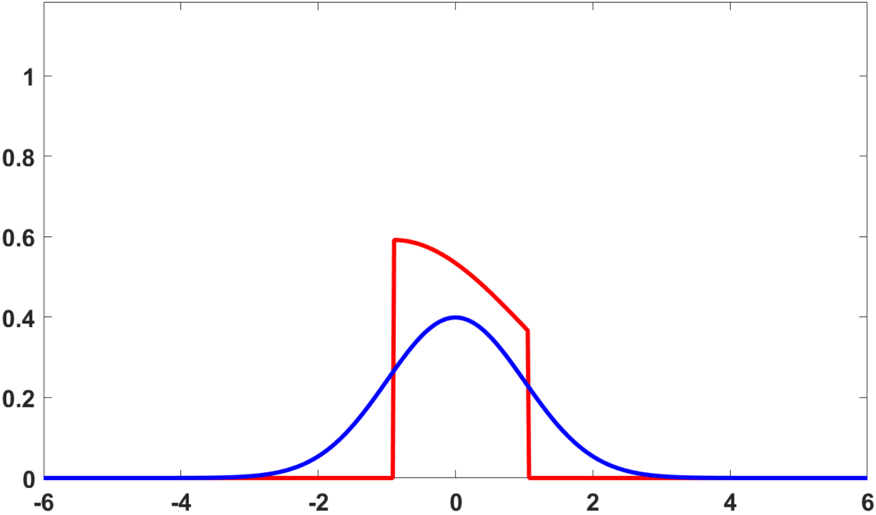
Multi-Stage Control with Finite-Rate Feedback

- $w_t \sim \mathcal{N}(0, W) \ast$ log-concave quantization error



Multi-Stage Control with Finite-Rate Feedback

- Convolution of log-concave functions is also **log-concave**!



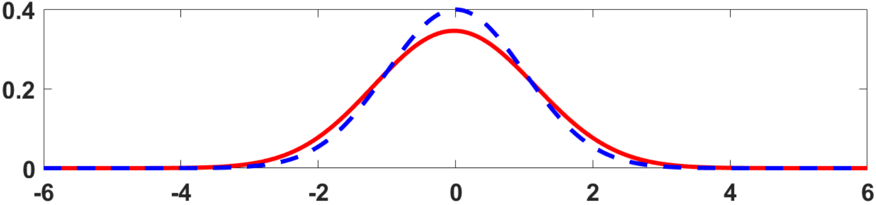
Multi-Stage Control with Finite-Rate Feedback

Resulting pdf (in red)

- Depends on cell index chosen in previous stage(s)
- Log-concave

Applying Lloyd-Max quantization in second stage is optimal!

- First-stage pdf (in blue) for comparison



Optimal Greedy Algorithm

- Lloyd-Max quantization minimizes squared error of that stage
- Lloyd-Max quantization = Optimal **greedy** algorithm
- But... It is not necessarily globally optimal...
- Current quantizer affects pdf of future stages
- Quantizer should be chosen according to the dynamic program (take into account the "cost-to-go")

Linear Quadratic Regulator (LQR) Example

- LQR setting with $x_0 \sim \mathcal{N}(0, X)$ and $\alpha = 1$:

$$\begin{cases} x_{t+1} & = x_t + u_t \\ y_t & = x_t \end{cases}$$

- Assume for simplicity we are interested in accumulated MMSE:

$$J = \sum_{t=1}^{T} \mathbb{E}\left[x_t^2\right] \triangleq \sum_{t=1}^{T} J_t$$
$$J_t \triangleq \mathbb{E}\left[x_t^2\right]$$

- In this case, clearly $u_1 = -\hat{x}_0$, $u_2 = -\left(\widehat{x_0 - \hat{x}_0}\right)$
- $\{u_t\}$ sequence refines the reconstruction of x_0 at every stage
- Equivalent to the *successive refinement* problem

Successive Refinement

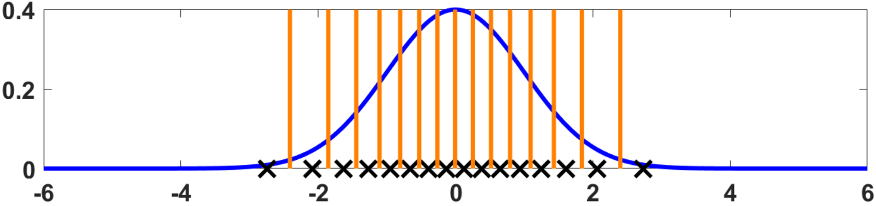
Successive refinement with encoding/decoding of long blocks [Equitz-Cover IT'91][Rimoldi IT'94]

- Optimal trade-off $(R_1, R_2) \leftrightarrow (J_1, J_2)$ is known
- J_2 is the same as if $R_1 + R_2$ was given to begin with (no J_1)

- But... **Optimal scalar quantizer for J_1 is not optimal for J_2**
- Tension between optimizing J_1 and J_2
- **Suboptimality of greedy algorithm** in LQR example [Fu AC'12]

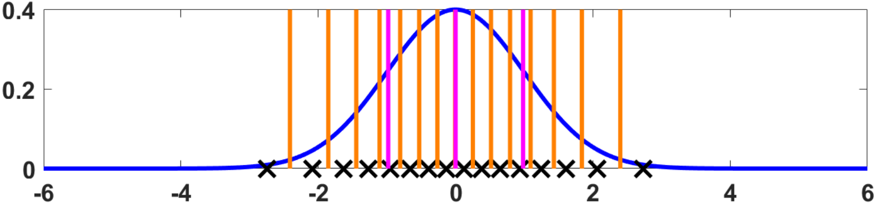
Successive Refinement: Example with $R = 2$

Lloyd-Max algorithm with $2^{2R} = 16$ quantization points:

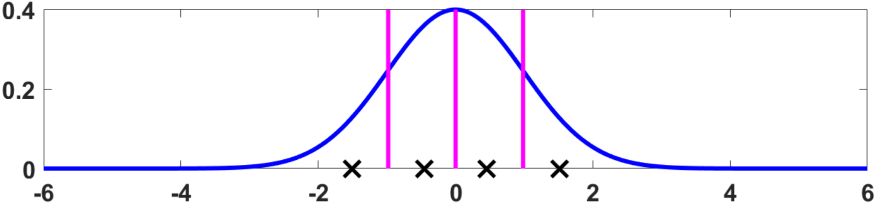


Successive Refinement: Example with $R = 2$

Lloyd-Max algorithm with $2^{2R} = 16$ quantization points:

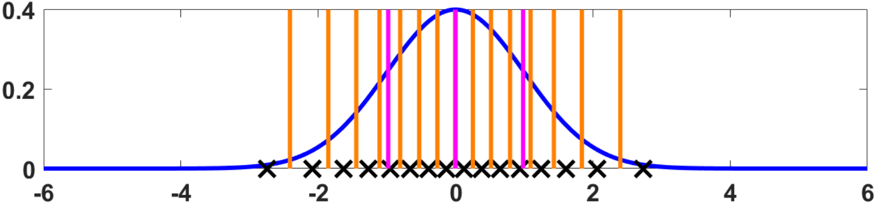


Lloyd-Max algorithm with $2^{R} = 4$ quantization points:

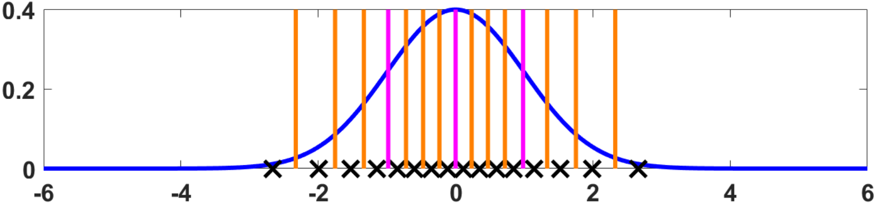


Successive Refinement: Example with $R = 2$

Lloyd-Max algorithm with $2^{2R} = 16$ quantization points:



Lloyd-Max algorithm ran for each cell with $2^R = 4$ points:



Optimal Scalar Successive Refinement

Optimal average-stage MMSE of scalar successive refinement [Dumitrescu-Wu IT'09]

- Generalized Lloyd-Max can be constructed [Brunk-Farvardin '96]
- Converges to **optimal average-stage MMSE**
- Extends Trushkin's conditions to successive refinement setting
- Conditions are satisfied for log-concave pdfs

LQR for $\alpha \neq 1$

- $J = J_1 + \alpha^2 J_2 + \cdots + \alpha^{2(T-1)} J_t$
- Adequate generalized LM algorithm can be constructed
- Converges to optimal weighted MSE J for log-concave pdfs

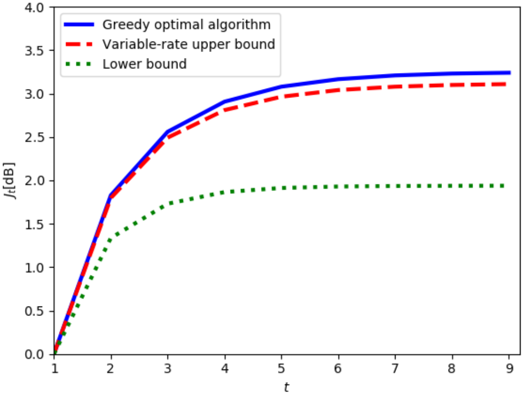
Performance

- $R = 1, W = 1$

LQR: $\alpha = 1.5$

t	Greedy	Optimal
1	1.0000	1.0000
2	1.8176	1.8177
3	2.4125	2.4126
4	2.8099	2.8064
5	3.0614	3.0514
6	3.2156	3.2001
7	3.3079	3.2877
8	3.3624	3.3381
9	3.3941	3.3688

LQG: $\alpha = 1.2$



High Resolution: Bennett's (Approx.) Optimal Quantizer

- Assume a large number of points
- Overload noise (noise outside dynamic range) is negligible
- Quantization points "can" be approximated by continuous pdf
- Optimal quantization points distribution $\propto f_X^{1/3}$
- Optimal distortion $= \frac{1}{12N^2} \|f_X\|_{1/3}$
- Under these assumptions $\rightarrow$ Successively refinable

No tensions between J_1 and J_2

Event-Triggering Control

- Treat silence (no transmission) as implicit information [Kofman-Braslavsky CDC'06][Khojasteh et al. Allerton'16, FrA16.1]
- Lloyd-Max like algorithm with extra (silent) bin & quant. point
- Probability of silence = constraint on the area of the silent bin
- $\Rightarrow$ Lloyd-Max with a constraint