Lossy Compression for Laplacian source

Hyeji Kim and Youngsuk Park

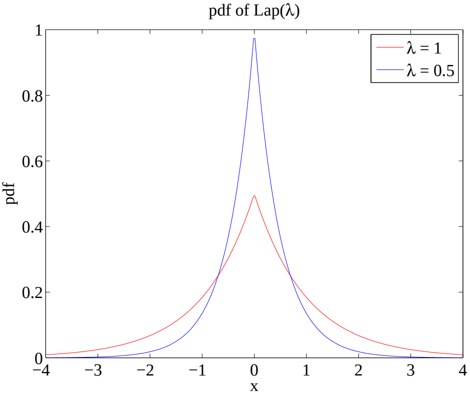
Stanford University

June 3, 2014

Laplacian source

- Laplacian source X^n, X_i is i.i.d. Laplacian(λ)

$$f_X(x) = \frac{1}{2\lambda} e^{-|x|/\lambda}$$



Lossy compression of Laplacian source

- Distortion: L1 norm

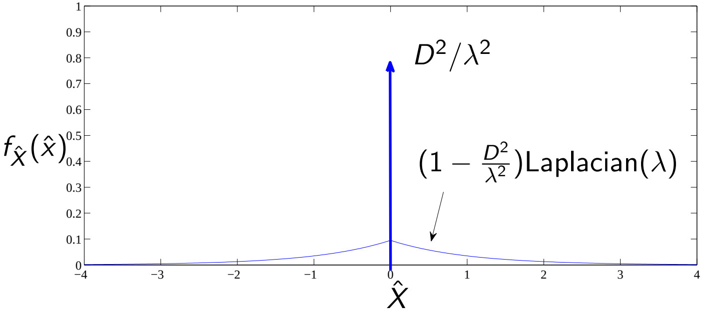
$$d(x^n, \hat{x}^n) = \frac{1}{n}\sum_{i=1}^{n} \|x_i - \hat{x}_i\|_1$$

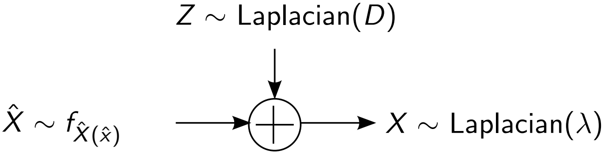
- Rate Distortion function

$$D(R) = \lambda e^{-R}$$
$$R(D) = log(\lambda/D)$$

Lossy compression of Laplacian source

- Rate distortion function $R(D) = \min_{p(\hat{x}|x):\ E[d(x,\hat{x})] \leq D} I(X;\hat{X})$





Rate distortion achieving scheme

- Source X^n; Target distortion D; Let $R = log(\lambda/D)$

Rate distortion achieving scheme

- Source X^n; Target distortion D; Let $R = log(\lambda/D)$
- Generation of codebook

$$\mathcal{C} = \{\hat{X}^n(1), \cdots, \hat{X}^n(e^{nR})\}, \text{ each } \hat{X}^n \sim \prod_{i=1}^{n} f(\hat{x}_i)$$

Rate distortion achieving scheme

- Source X^n; Target distortion D; Let $R = log(\lambda/D)$
- Generation of codebook

$$\mathcal{C} = \{\hat{X}^n(1), \cdots, \hat{X}^n(e^{nR})\}, \text{ each } \hat{X}^n \sim \prod_{i=1}^{n} f(\hat{x}_i)$$

- Encoding: Encode X^n by

$$w = \arg \min_{w' \in [1:e^{nR}]} \|X^n - \hat{X}^n(w')\|_1$$

Rate distortion achieving scheme

- Source X^n; Target distortion D; Let $R = log(\lambda/D)$
- Generation of codebook

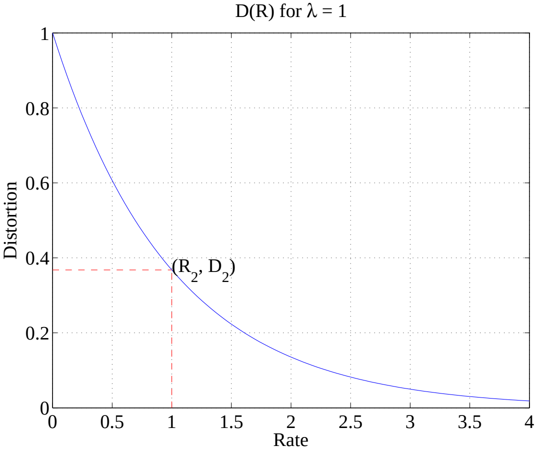
$$\mathcal{C} = \{\hat{X}^n(1), \cdots, \hat{X}^n(e^{nR})\}, \text{ each } \hat{X}^n \sim \prod_{i=1}^{n} f(\hat{x}_i)$$

- Encoding: Encode X^n by

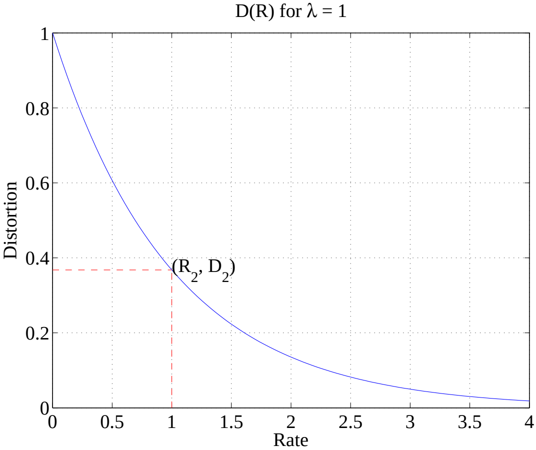
$$w = \arg \min_{w' \in [1:e^{nR}]} \|X^n - \hat{X}^n(w')\|_1$$

- Decoding: The reproduced sequence is $\hat{X}^n(w)$

Rate Distortion achieving scheme



Rate Distortion achieving scheme



- Codebook size to achieve distortion D_2 is e^{nR_2}

Successive Refinement

- Can we reduce the codebook size?

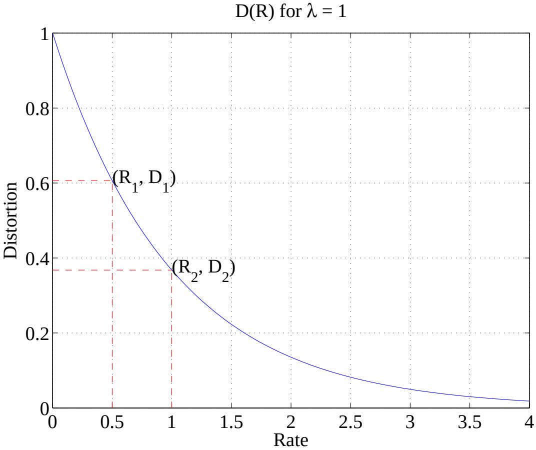
Successive Refinement

- Can we reduce the codebook size?

Successive Refinement

- Can we reduce the codebook size?

Yes! Successive refinement



Successive Refinement

- Step 1: Optimal lossy compression for Laplacian (λ) with target distortion D_1 at rate $R_1 = R_2/2$

Successive Refinement

- Step 1: Optimal lossy compression for Laplacian (λ) with target distortion D_1 at rate $R_1 = R_2/2$
  - Receiver reproduces $\hat{X}^n(w_1)$

Successive Refinement

- Step 1: Optimal lossy compression for Laplacian (λ) with target distortion D_1 at rate $R_1 = R_2/2$
  - Receiver reproduces $\hat{X}^n(w_1)$
  - Key observation: $X^n - \hat{X}^n(w_1)$ is Laplacian (D_1)

Successive Refinement

- Step 1: Optimal lossy compression for Laplacian (λ) with target distortion D_1 at rate $R_1 = R_2/2$
  - Receiver reproduces $\hat{X}^n(w_1)$
  - Key observation: $X^n - \hat{X}^n(w_1)$ is Laplacian (D_1)
- Step 2: Optimal lossy compression for Laplacian (D_1) with target distortion D_2 at rate $R_1 = R_2/2$

Successive Refinement

- Step 1: Optimal lossy compression for Laplacian (λ) with target distortion D_1 at rate $R_1 = R_2/2$
  - Receiver reproduces $\hat{X}^n(w_1)$
  - Key observation: $X^n - \hat{X}^n(w_1)$ is Laplacian (D_1)
- Step 2: Optimal lossy compression for Laplacian (D_1) with target distortion D_2 at rate $R_1 = R_2/2$
  - Receiver reproduces $\hat{X}^n(w_1) + \hat{X}^n(w_2)$

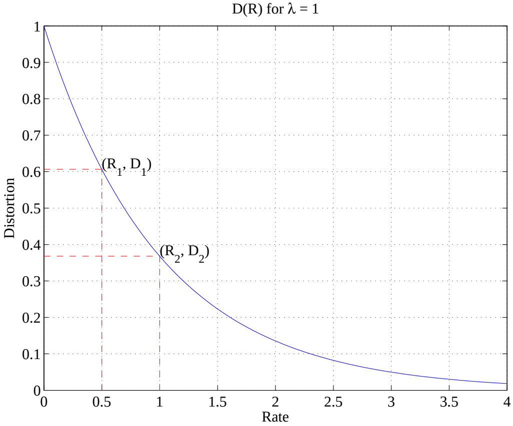
Successive Refinement

- Step 1: Optimal lossy compression for Laplacian (λ) with target distortion D_1 at rate $R_1 = R_2/2$
  - Receiver reproduces $\hat{X}^n(w_1)$
  - Key observation: $X^n - \hat{X}^n(w_1)$ is Laplacian (D_1)
- Step 2: Optimal lossy compression for Laplacian (D_1) with target distortion D_2 at rate $R_1 = R_2/2$
  - Receiver reproduces $\hat{X}^n(w_1) + \hat{X}^n(w_2)$
  - Key observation: $X^n - \hat{X}^n(w_1) - \hat{X}^n(w_2)$ is Laplacian (D_2)

Successive Refinement

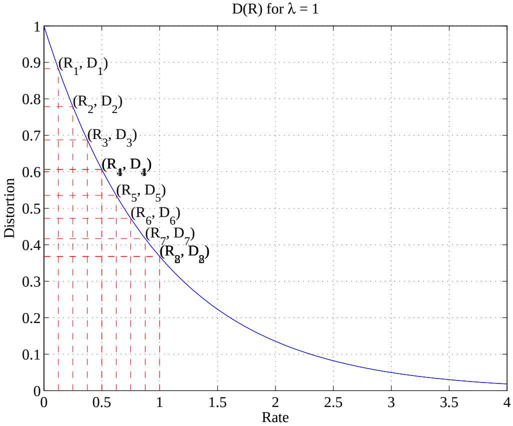
- Step 1: Optimal lossy compression for Laplacian (λ) with target distortion D_1 at rate $R_1 = R_2/2$
  - Receiver reproduces $\hat{X}^n(w_1)$
  - Key observation: $X^n - \hat{X}^n(w_1)$ is Laplacian (D_1)
- Step 2: Optimal lossy compression for Laplacian (D_1) with target distortion D_2 at rate $R_1 = R_2/2$
  - Receiver reproduces $\hat{X}^n(w_1) + \hat{X}^n(w_2)$
  - Key observation: $X^n - \hat{X}^n(w_1) - \hat{X}^n(w_2)$ is Laplacian (D_2)
- Codebook size: $2e^{nR_2/2} = e^{nR_2/2}$(step 1) $+e^{nR_2/2}$(step 2)

Successive Refinement



- Codebook size: $2e^{nR_2/2} = e^{nR_2/2}(\text{step 1}) + e^{nR_2/2}(\text{step 2})$

Successive Refinement



- Codebook size: $Le^{nR_2/L} << e^{nR_2}$

Generalized Successive Refinement

- For any fixed L, codebook size $= Le^{nR/L}$

Generalized Successive Refinement

- For any fixed L, codebook size $= Le^{nR/L}$
- Can we reduce codebook size further?

Generalized Successive Refinement

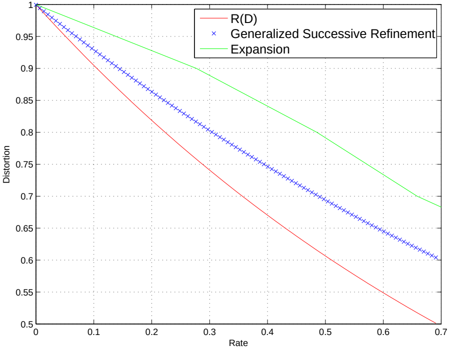
- For any fixed L, codebook size $= Le^{nR/L}$
- Can we reduce codebook size further?
- Yes, L_n increases with n

Generalized Successive Refinement

- For any fixed L, codebook size $= Le^{nR/L}$
- Can we reduce codebook size further?
- Yes, L_n increases with n
- We choose $L_n = nR/\log n$
  Rate increment $R/L_n = \log n/n$
  Sub codebook size $e^{nR/L_n} = n$
  Number of total codewords $L_n e^{nR/L_n} = n^2 R/\log n << Le^{nR/L}$

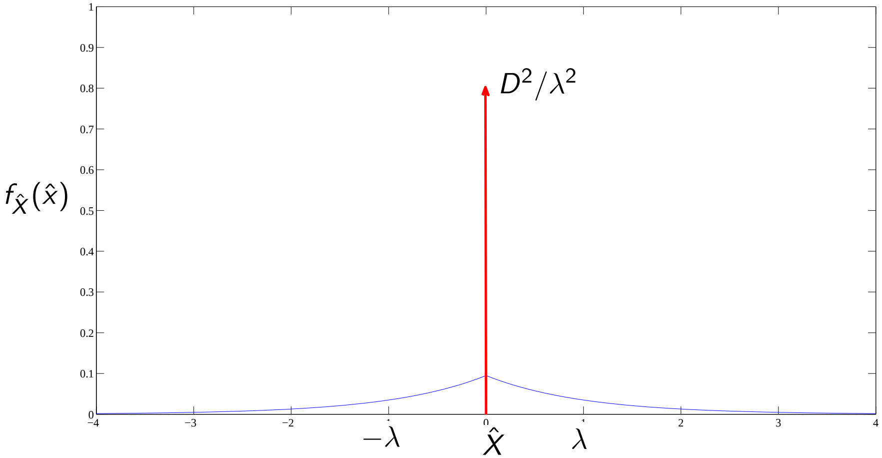
Performance of generalized successive refinement

- n=1000, subcodebook size=1000, averaged over 1000 experiments

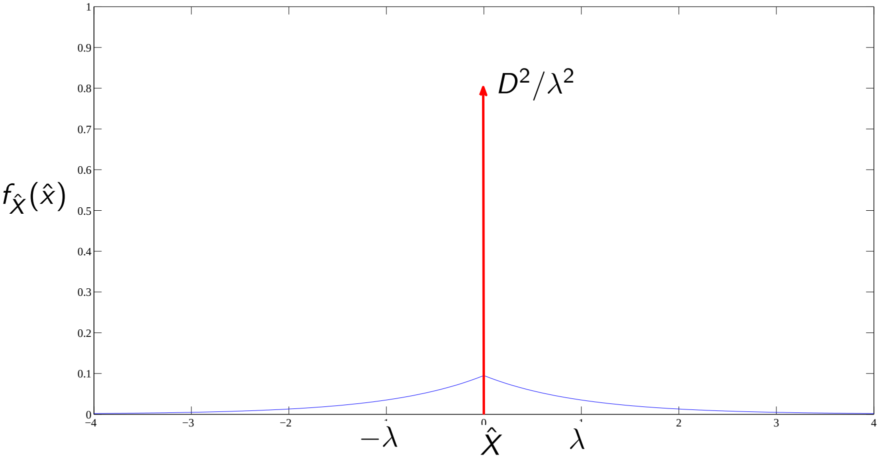


- Better than expansion coding[Si, Koyluoglu, Vishwanath 2013]
- Rateless and sequential!

Generalized successive refinement with finite alphabet

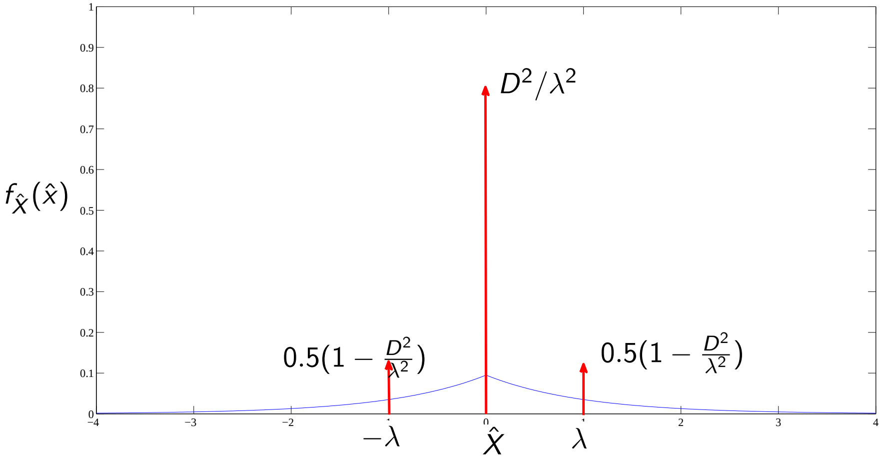


Generalized successive refinement with finite alphabet



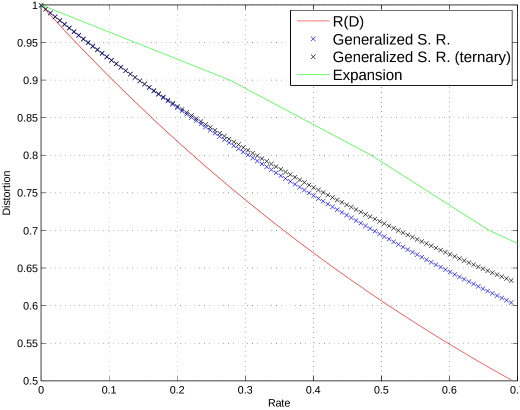
- Codewords $\hat{X}^n$ has infinite alphabet size

Generalized successive refinement with finite alphabet



- Codewords $\hat{X}^n$ has finite alphabet size

Performance



Summary

- We generalized successive refinement, and designed a practical code for Laplacian source
- Benefits of generalized successive refinement
  - Small codebook size, low complexity
  - Rateless and sequential

Ongoing work

- Focus on rateless and sequential scheme
  - Sending max/min indices
  - Applying our schemes on non-Laplacian sources, e.g. Gaussian

Special thanks to

Albert No