COMUNICATION BY REGRESSION: Sparse Superposition Codes

Andrew Barron
Department of Statistics
Yale University

Joint work with Sanghee Cho, Antony Joseph

Statistics Department Seminar
Cambridge University
November 14, 2013

Quest for Provably Practical and Reliable High Rate Communication

- The Channel Communication Problem
- Gaussian Channel
- History of Methods
- Communication by Regression
- Sparse Superposition Coding
- Adaptive Successive Decoding
- Rate, Reliability, and Computational Complexity
- Distributional Analysis
- Simulations

Shannon Formulation

- Input bits: $U = (U_1, U_2, \ldots\ldots, U_K)$ indep Bern(1/2)

$$\downarrow$$

- Encoded: $x = (x_1, x_2, \ldots, x_n)$

$$\downarrow$$

- Channel: $p(y|x)$

$$\downarrow$$

- Received: $Y = (Y_1, Y_2, \ldots, Y_n)$

$$\downarrow$$

- Decoded: $\hat{U} = (\hat{U}_1, \hat{U}_2, \ldots\ldots, \hat{U}_K)$

- Rate: $R = \frac{K}{n}$ Capacity $C = \max_{P_X} I(X;Y)$

- Reliability: Want small Prob$\{\hat{U} \neq U\}$
  or reliably small fraction of errors

Gaussian Noise Channel

- Input bits: $U = (U_1, U_2, \ldots\ldots, U_K)$

$$\downarrow$$

- Encoded: $x = (x_1, x_2, \ldots, x_n)$ $\quad \frac{1}{n}\sum_{i=1}^{n} x_i^2 \cong P$

$$\downarrow$$

- Channel: $p(y|x)$ $\quad Y = x(U) + \varepsilon\,,\ \varepsilon \sim N(0, \sigma^2 I)$

$$snr = P/\sigma^2$$

$$\downarrow$$

- Received: $Y = (Y_1, Y_2, \ldots, Y_n)$

$$\downarrow$$

- Decoded: $\hat{U} = (\hat{U}_1, \hat{U}_2, \ldots\ldots, \hat{U}_K)$
- Rate: $R = \frac{K}{n}$ $\quad$ Capacity $C = \frac{1}{2}\log(1 + snr)$
- Reliability: Want small Prob$\{\hat{U} \neq U\}$
  or reliably small fraction of errors

Shannon Theory

- Channel Capacity:
  Supremum of rates R such that reliable communication is possible, with arbitrarily small error probability

- Information Capacity: $C = \max_{P_X} I(X; Y)$

  Where $I(X; Y)$ is the Shannon information, also known as the Kullback divergence between $P_{X,Y}$ and $P_X \times P_Y$

- Shannon Channel Capacity Theorem:
  The supremum of achievable communication rates R equals the information capacity C

- Books:

  Shannon (49), Gallager (68), Cover & Thomas (06)

The Gaussian Noise Model

The Gaussian noise channel is the basic model for

- wireless communication

  radio, cell phones, television, satellite, space

- wired communication

  internet, telephone, cable

Shannon Theory meets Coding Practice

- Forney and Ungerboeck 1998 review:
  - modulation, shaping and coding for the Gaussian channel
- Richardson & Urbanke 2008, state of the art:
  - Empirically good LDPC and turbo codes with fast encoding and decoding based on Bayesian belief networks
  - New spatial coupling techniques, Urbanke 2013
  - Proof of rates up to capacity in some cases
- Arikan 2009, Arikan and Teletar 2009 polar codes:
  - Adapted to Gaussian channel (Abbe and Barron 2011)
- Tropp 2006,2008 codes from compressed sensing and related sparse signal recovery work:
  - Wainwright; Fletcher, Rangan, Goyal; Zhang; others
  - Donoho,Montenari, et al 2012, role of spatial coupling
  - ℓ_1-constrained least squares practical, has positive rate
  - but not capacity achieving
- Knowledge of above not necessary to follow presentation

Sparse Superposition Code

- Input bits: $U = (U_1 \ldots\ldots\ldots\ldots U_K)$
- Coefficients: $\beta = (00 * 0000000000 * 00 \ldots 0 * 000000)^T$
- Sparsity: L entries non-zero out of N
- Matrix: X, n by N, all entries indep Normal$(0, 1)$
- Codeword: $X\beta$, superposition of a subset of columns
- Receive: $Y = X\beta + \varepsilon$, a statistical linear model
- Decode: $\hat{\beta}$ and $\hat{U}$ from X, Y

Sparse Superposition Code

- Input bits: $U = (U_1 \ldots\ldots\ldots\ldots U_K)$
- Coefficients: $\beta = (00 * 0000000000 * 00 \ldots 0 * 000000)^T$
- Sparsity: L entries non-zero out of N
- Matrix: X, n by N, all entries indep Normal$(0, 1)$
- Codeword: $X\beta$, superposition of a subset of columns
- Receive: $Y = X\beta + \varepsilon$, a statistical linear model
- Decode: $\hat{\beta}$ and $\hat{U}$ from X, Y

Sparse Superposition Code

- Input bits: $U = (U_1 \ldots\ldots\ldots\ldots U_K)$
- Coefficients: $\beta = (00 * 0000000000 * 00 \ldots 0 * 000000)^T$
- Sparsity: L entries non-zero out of N
- Matrix: X, n by N, all entries indep Normal$(0, 1)$
- Codeword: $X\beta$, superposition of a subset of columns
- Receive: $Y = X\beta + \varepsilon$
- Decode: $\hat{\beta}$ and $\hat{U}$ from X, Y
- Rate: $R = \frac{K}{n}$ from $K = \log \binom{N}{L}$, near $L \log \left(\frac{N}{L} e\right)$

Sparse Superposition Code

- Input bits: $U = (U_1 \ldots\ldots\ldots\ldots U_K)$
- Coefficients: $\beta = (00 * 0000000000 * 00 \ldots 0 * 000000)^T$
- Sparsity: L entries non-zero out of N
- Matrix: X, n by N, all entries indep Normal$(0, 1)$
- Codeword: $X\beta$, superposition of a subset of columns
- Receive: $Y = X\beta + \varepsilon$
- Decode: $\hat{\beta}$ and $\hat{U}$ from X, Y
- Rate: $R = \frac{K}{n}$ from $K = \log \binom{N}{L}$
- Reliability: exponentially small probability of error

Want reliability with rate up to capacity

Partitioned Superposition Code

- Input bits: $U = (U_1 \ldots, \ \ldots, \ \ldots, \ \ldots U_K)$
- Coefficients: $\beta = (00 * 00000, \ 00000 * 00, \ \ldots, \ 0 * 000000)$
- Sparsity: L sections, each of size $M = N/L$, a power of 2
  1 non-zero entry in each section
- Indices of nonzeros: $(j_1, j_2, \ldots, j_L)$ specified by U segments
- Matrix: X, n by N, splits into L sections
- Codeword: $X\beta$, superposition of columns, one from each
- Receive: $Y = X\beta + \varepsilon$
- Decode: $\hat{\beta}$ and $\hat{U}$
- Rate: $R = \frac{K}{n}$ from $K = L \log \frac{N}{L} = L \log M$

Partitioned Superposition Code

- Input bits: $U = (U_1 \ldots, \ \ldots, \ \ldots, \ \ldots U_K)$
- Coefficients: $\beta = (00 * 00000, \ 00000 * 00, \ \ldots, \ 0 * 000000)$
- Sparsity: L sections, each of size $M = N/L$, a power of 2
  1 non-zero entry in each section
- Indices of nonzeros: $(j_1, j_2, \ldots, j_L)$ specified by U segments
- Matrix: X, n by N, splits into L sections
- Codeword: $X\beta$, superposition of columns, one from each
- Receive: $Y = X\beta + \varepsilon$
- Decode: $\hat{\beta}$ and $\hat{U}$
- Rate: $R = \frac{K}{n}$ from $K = L \log \frac{N}{L} = L \log M$
- Reliability: small Prob$\{$*Fraction* $\hat{\beta}$ *mistakes* $\geq \alpha\}$, small α

Is it reliable with rate up to capacity?

Partitioned Superposition Code

- Input bits: $U = (U_1 \ldots, \ \ldots, \ \ldots, \ \ldots U_K)$
- Coefficients: $\beta = (00 * 00000, \ 00000 * 00, \ \ldots, \ 0 * 000000)$
- Sparsity: L sections, each of size $M = N/L$, a power of 2
  1 non-zero entry in each section
- Indices of nonzeros: $(j_1, j_2, \ldots, j_L)$ specified by U segments
- Matrix: X, n by N, splits into L sections
- Codeword: $X\beta$, superposition of columns, one from each
- Receive: $Y = X\beta + \varepsilon$
- Decode: $\hat{\beta}$ and $\hat{U}$
- Rate: $R = \frac{K}{n}$ from $K = L \log \frac{N}{L} = L \log M$
- Ultra-sparse case: Impractical $M = 2^{nR/L}$ with L constant
  (reliable at all rates $R < C$: Cover 1972,1980)
- Moderately-sparse: Practical $M = n$ with $L = nR/\log n$
  (Still reliable at all $R < C$)

Partitioned Superposition Code

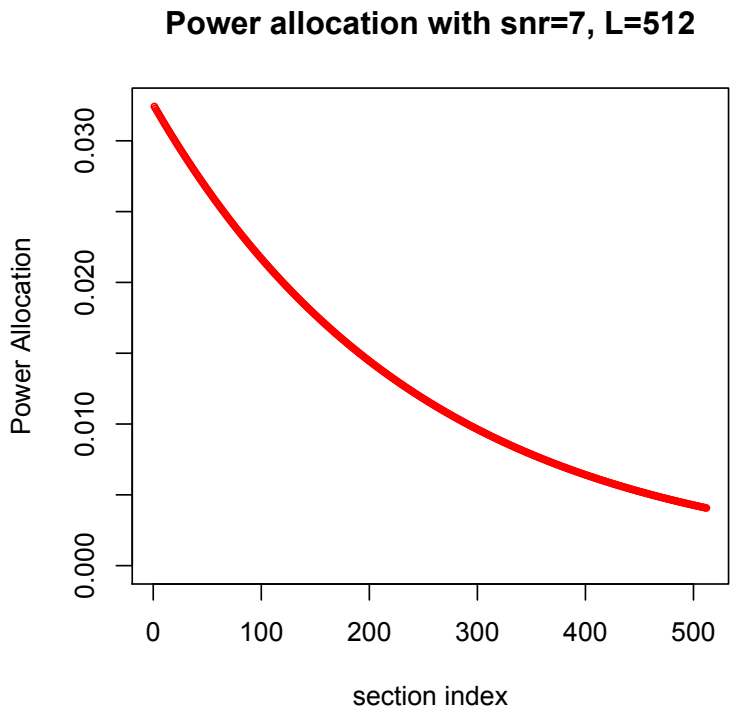
- Input bits: $U = (U_1 \ldots, \ \ldots, \ \ldots, \ \ldots U_K)$
- Coefficients: $\beta = (00 * 00000, \ 00000 * 00, \ \ldots, \ 0 * 000000)$
- Sparsity: L sections, each of size $M = N/L$, a power of 2
  1 non-zero entry in each section
- Indices of nonzeros: $(j_1, j_2, \ldots, j_L)$ specified by U segments
- Matrix: X, n by N, splits into L sections
- Codeword: $X\beta$, superposition of columns, one from each
- Receive: $Y = X\beta + \varepsilon$
- Decode: $\hat{\beta}$ and $\hat{U}$
- Rate: $R = \frac{K}{n}$ from $K = L \log \frac{N}{L} = L \log M$
- Reliability: small Prob$\{$*Fraction mistakes* $\geq \alpha\}$ small α
- Outer RS code: rate $1-\alpha$, corrects remaining mistakes
- Overall rate: $R_{tot} = (1-\alpha)R$
- Overall rate: up to capacity

Power Allocation

- Coefficients: $\beta = (00{*}00000,\ 00000{*}00, \ldots, 0{*}000000)$
- Indices of nonzeros: $sent = (j_1, j_2, \ldots, j_L)$
- Coeff. values: $\beta_{j_\ell} = \sqrt{P_\ell}$ for $\ell = 1, 2, \ldots, L$
- Power control: $\sum_{\ell=1}^{L} P_\ell = P$
- Codewords: $X\beta$, have average power P
- Power Allocations
  - Constant power: $P_\ell = P/L$
  - Variable power: P_ℓ proportional to $e^{-2C\,\ell/L}$

Variable Power Allocation

- Power control: $\sum_{\ell=1}^{L} P_\ell = P \qquad \|\beta\|^2 = P$
- Variable power: P_ℓ proportional to $e^{-2C\ell/L}$ for $\ell = 1, \ldots, L$



Variable Power Allocation

- Power control: $\sum_{\ell=1}^{L} P_\ell = P \qquad\qquad \|\beta\|^2 = P$
- Variable power: P_ℓ proportional to $e^{-2C\ell/L}$ for $\ell = 1, \ldots, L$
- Successive decoding motivation
- Incremental capacity

$$\frac{1}{2}\log\left(1 + \frac{P_\ell}{\sigma^2 + P_{\ell+1} + \cdots + P_L}\right) = \frac{C}{L}$$

matching the section rate

$$\frac{R}{L} = \frac{\log M}{n}$$

Adaptive Successive Decoder

Decoding Steps (with thresholding)

- Start: [Step 1]
  - Compute the inner product of Y with each column of X
  - See which are above a threshold
  - Form initial fit as weighted sum of columns above threshold
- Iterate: [Step $k \geq 2$]
  - Compute the inner product of residuals $Y - Fit_{k-1}$ with each remaining column of X
  - Standardize by dividing by $\| Y - Fit_{k-1} \|$
  - See which are above the threshold $\sqrt{2 \log M} + a$
  - Add these columns to the fit
- Stop:
  - At Step $k = 1 + snr \log M$, or
  - if there are no additional inner products above threshold

Complexity of Adaptive Successive Decoder

Complexity in parallel pipelined implementation

- Space: (use $k = snr \log M$ copies of the n by N dictionary)
  - $knN = snr\, C\, M\, n^2$ memory positions
  - kN multiplier/accumulators and comparators
- Time: $O(1)$ per received Y symbol

Adaptive Successive Decoder

Decoding Steps (with iteratively optimal statistics)

- Start: [Step 1]
  - Compute the inner product of Y with each column of X
  - Form initial fit
- Iterate: [Step $k \geq 2$]
  - Compute inner product of residuals $Y - Fit_{k-1}$ with each X_j.
  - Adjusted form: $\text{stat}_{k,j}$ equals $(Y - X\hat{\beta}_{k-1,-j})^T X_j$
  - Standardize by dividing it by $\|Y - X\hat{\beta}_{k-1}\|$.
  - Form the new fit

$$\hat{\beta}_{k,j} = \sqrt{P_\ell}\,\hat{w}_j(\alpha) = \sqrt{P_\ell}\,\frac{e^{\alpha\,\text{stat}_{k,j}}}{\sum_{j\in sec_\ell} e^{\alpha\,\text{stat}_{k,j}}}$$

- Stop:
  - When estimated success stops increasing
  - At Step $k = O(\log M)$

Iteratively Bayes optimal $\hat{\beta}_k$

With prior $j_\ell \sim \mathit{Unif}$ on sec_ℓ, the Bayes estimate based on stat_k

$$\hat{\beta}_k = \mathbb{E}[\beta | \mathit{stat}_k]$$

has representation $\hat{\beta}_{k,j} = \sqrt{P_\ell}\, \hat{w}_{k,j}$ with

$$\hat{w}_{k,j} = \mathit{Prob}\{j_\ell = j | \mathit{stat}_k\}.$$

Iteratively Bayes optimal $\hat{\beta}_k$

With prior $j_\ell \sim Unif$ on sec_ℓ, the Bayes estimate based on $stat_k$

$$\hat{\beta}_k = \mathbb{E}[\beta | stat_k]$$

has representation $\hat{\beta}_{k,j} = \sqrt{P_\ell}\, \hat{w}_{k,j}$ with

$$\hat{w}_{k,j} = Prob\{j_\ell = j | stat_k\}.$$

Here, when the $stat_{k,j}$ are independent $N(\alpha_{\ell,k} 1_{\{j=j_\ell\}}, 1)$, we have the logit representation $\hat{w}_{k,j} = \hat{w}(\alpha_{\ell,k})$ where

$$\hat{w}_{k,j}(\alpha) = \frac{e^{\alpha\, stat_{k,j}}}{\sum_{j \in sec_\ell} e^{\alpha\, stat_{k,j}}}.$$

Iteratively Bayes optimal $\hat{\beta}_k$

With prior $j_\ell \sim Unif$ on sec_ℓ, the Bayes estimate based on $stat_k$

$$\hat{\beta}_k = \mathbb{E}[\beta | stat_k]$$

has representation $\hat{\beta}_{k,j} = \sqrt{P_\ell}\, \hat{w}_{k,j}$ with

$$\hat{w}_{k,j} = Prob\{j_\ell = j | stat_k\}.$$

Here, when the $stat_{k,j}$ are independent $N(\alpha_{\ell,k} 1_{\{j=j_\ell\}}, 1)$, we have the logit representation $\hat{w}_{k,j} = \hat{w}(\alpha_{\ell,k})$ where

$$\hat{w}_{k,j}(\alpha) = \frac{e^{\alpha\, stat_{k,j}}}{\sum_{j \in sec_\ell} e^{\alpha\, stat_{k,j}}}.$$

Recall $\mathrm{stat}_{k,j}$ is standardized inner product of residuals with X_j

$$\mathrm{stat}_{k,j} = \frac{(Y - X\hat{\beta}_{k-1,-j})^T X_j}{\|Y - X\hat{\beta}_{k-1}\|}$$

Distributional Analysis

- Approximate distribution of these statistics: independent standard normal, shifted for terms sent

$$\text{stat}_{k,j} = \alpha_{\ell,x_{k-1}} 1_{\{j \text{ sent}\}} + Z_{k,j}$$

where

$$\alpha = \alpha_{\ell,x} = \sqrt{\frac{nP_\ell}{\sigma^2 + P(1-x)}}$$

- Here

$$E\|\hat{\beta}_{k-1} - \beta\|^2 = P\,(1 - x_{k-1})$$

- Update rule $x_k = g(x_{k-1})$ where

$$g(x) = \sum_{\ell=1}^{L} (P_\ell/P) E[w_{j_\ell}(\alpha_{\ell,x})].$$

Distributional Analysis

- Approximate distribution of these statistics: independent standard normal, shifted for terms sent

$$\text{stat}_{k,j} = \alpha_{\ell,x_{k-1}} 1_{\{j \text{ sent}\}} + Z_{k,j}$$

where

$$\alpha = \alpha_{\ell,x} = \sqrt{\frac{nP_\ell}{\sigma^2 + P(1-x)}}$$

- Here

$$E\|\hat{\beta}_{k-1} - \beta\|^2 = P\,(1 - x_{k-1})$$

- Update rule $x_k = g(x_{k-1})$ where

$$g(x) = \sum_{\ell=1}^{L} (P_\ell/P) E[w_{j_\ell}(\alpha_{\ell,x})].$$

Success Rate Update Rule

- Update rule $x_k = g(x_{k-1})$ where

$$g(x) = \sum_{\ell=1}^{L} (P_\ell / P) E[w_{j_\ell}(\alpha_{\ell,x})]$$

- this is the success rate update function expressed as a weighted sum of the posterior prob of the term sent

Success Rate Update Rule

- Update rule $x_k = g(x_{k-1})$ where

$$g(x) = \sum_{\ell=1}^{L} (P_\ell / P) E[w_{j_\ell}(\alpha_{\ell,x})]$$

- this is the success rate update function expressed as a weighted sum of the posterior prob of the term sent
- Empirical success rate

$$x_k^* = \sum_{\ell=1}^{L} (P_\ell / P) w_{j_\ell}(\alpha_{\ell,x_k})$$

Success Rate Update Rule

- Update rule $x_k = g(x_{k-1})$ where

$$g(x) = \sum_{\ell=1}^{L} (P_\ell / P) E[w_{j_\ell}(\alpha_{\ell,x})]$$

- this is the success rate update function expressed as a weighted sum of the posterior prob of the term sent
- Empirical success rate

$$x_k^* = \sum_{\ell=1}^{L} (P_\ell / P) w_{j_\ell}(\alpha_{\ell,x_k})$$

- Empirical estimated success rate

$$\hat{x}_k = \sum_{\ell=1}^{L} (P_\ell / P) \sum_{j \in sec_\ell} [w_j(\alpha_{\ell,x_k})]^2$$

Decoding progression

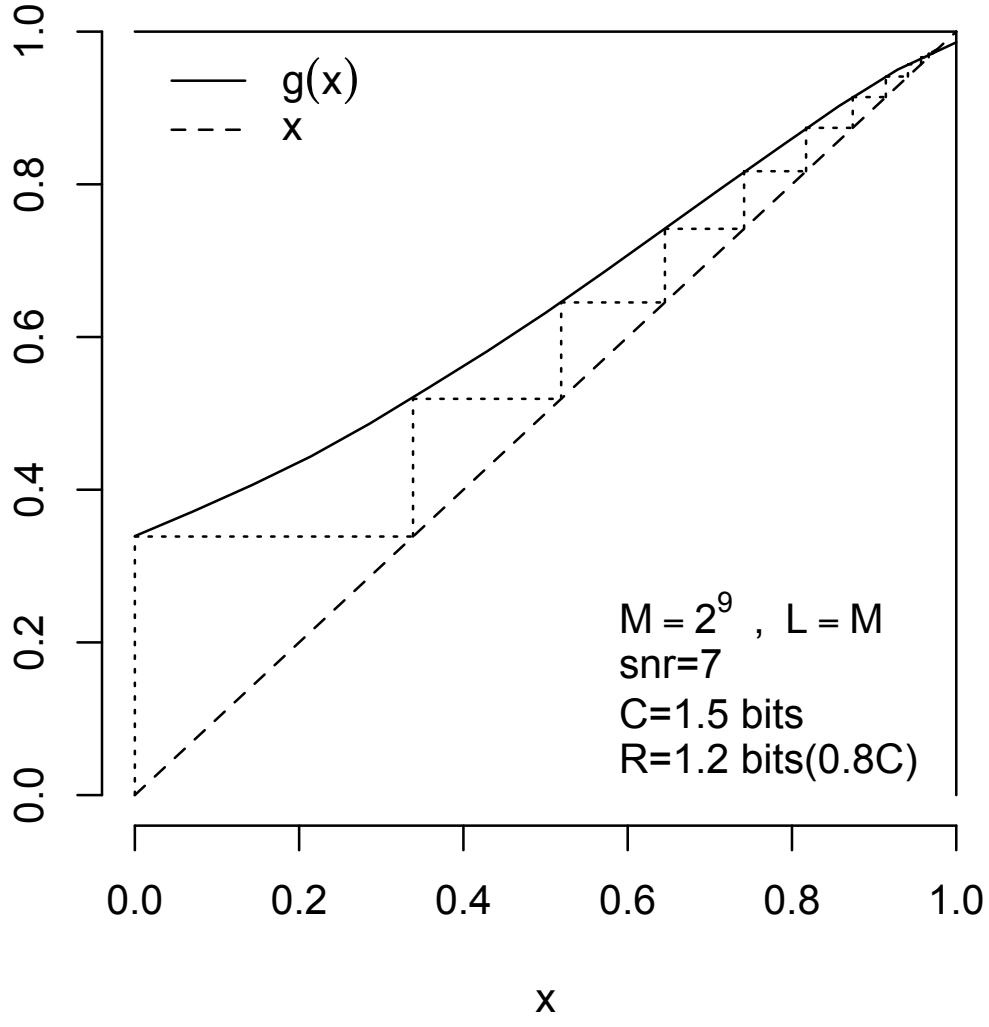


Figure : Plot of $g(x)$ and the sequence x_k.

Update fuctions

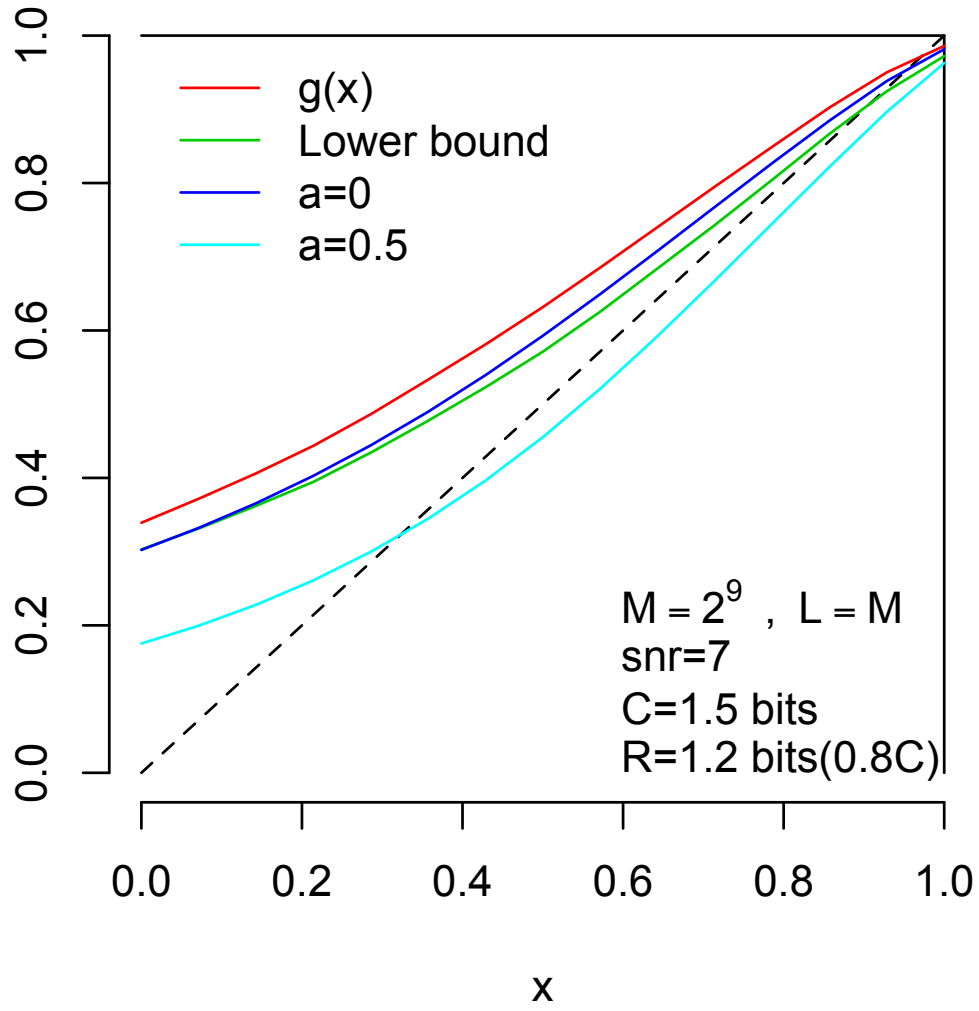


Figure : Comparison of update functions. Blue and light blue lines indicates $\{0, 1\}$ decision using the threshold $\tau = \sqrt{2logM} + a$ with respect to the value a as indicated.

Success Progression plots

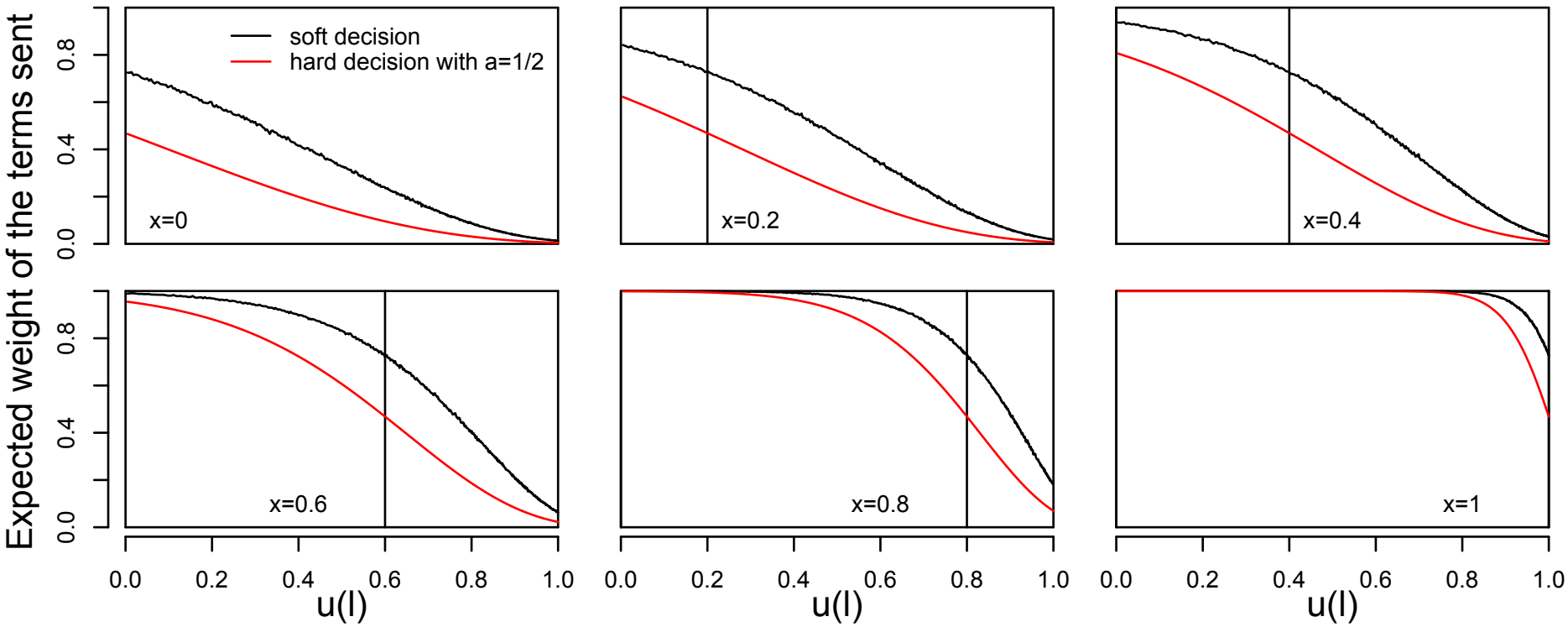


Figure : Progression plots : $M = 2^9$, $L = M$, $C = 1.5$ bits and $R = 0.8C$. We used Monte Carlo simulation with replicate size 10000. The horizontal axis depicts $u(\ell) = 1 - e^{-2C\ell/L}$ which is an increasing function of ℓ. Area under curve equals g(x).

Rate and Reliability

Result for Optimal ML Decoder [Joseph and B. 2012], with outer RS decoder, and with equal power allowed across the sections

- Prob error exponentially small in n for all $R < C$

$$\text{Prob}\{Error\} \leq e^{-n(C-R)^2/2V}$$

- In agreement with the Shannon-Gallager exponent of optimal code, though with a suboptimal constant V depending on the *snr*

Rate and Reliability of Fast Superposition Decoder

Practical: Adaptive Successive Decoder, with outer RS code.

- prob error exponentially small in $n/(\log M)$ for $R < C$
- Value C_M approaching capacity

$$C_M = \frac{C}{1 + c_1 / \log M}$$

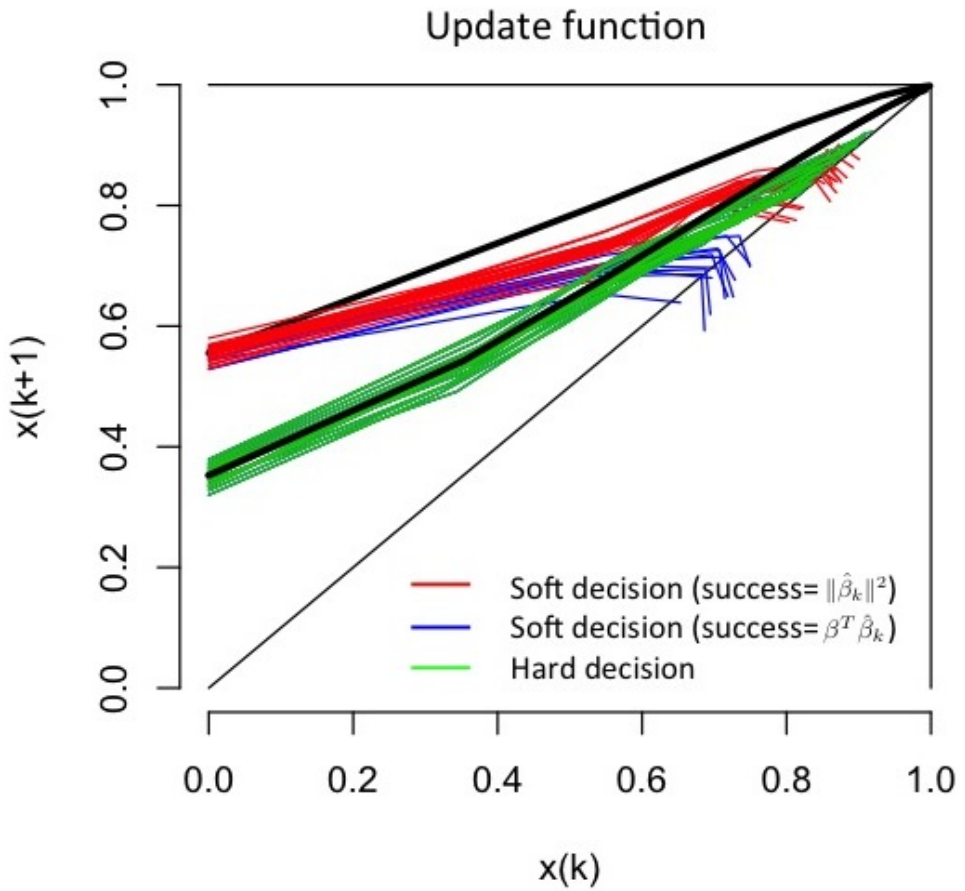
- Probability error exponentially small in L for $R < C_M$

$$\text{Prob}\{\textit{Error}\} \leq e^{-L(C_M - R)^2 c_2}$$

- Improves to $e^{-c_3 L (C_M - R)^2 (\log M)^{0.5}}$ using a Bernstein bound.

Simulation

Figure : $L = 512$, $M = 64$, $snr = 15$, $C = 2$ *bits*, $R = 1$ *bits*, $n = 3072$. Ran 20 trials for each method. Green lines for hard thresholding; blue and red for soft decision decoder.



Framework of Iterative Statistics

For $k \geq 1$

- Codeword fits: $F_k = X\hat{\beta}_k$
- Vector of statistics: $stat_k =$ function of $(X, Y, F_1, \ldots, F_k)$
- e.g. $stat_{k,j}$ proportional to $X_j^T(Y - F_k)$
- Update $\hat{\beta}_{k+1}$ as a function of $stat_k$

Framework of Iterative Statistics

For $k \geq 1$

- Codeword fits: $F_k = X\hat{\beta}_k$
- Vector of statistics: $stat_k =$ function of $(X, Y, F_1, \ldots, F_k)$
- e.g. $stat_{k,j}$ proportional to $X_j^T(Y - F_k)$
- Update $\hat{\beta}_{k+1}$ as a function of $stat_k$
  - Thresholding: Adaptive Successive Decoder
    $\hat{\beta}_{k+1,j} = \sqrt{P_\ell}\, 1_{\{stat_{k,j} > thres\}}$
  - Soft decision:
    $\hat{\beta}_{k+1,j} = \mathbb{E}[\beta_j | stat_k] = \sqrt{P_\ell}\, \hat{w}_{k,j}$
    with thresholding on the last step

Orthogonal Components

- Codeword fits: $F_k = X\hat{\beta}_k$
- Orthogonalization : Let $G_0 = Y$ and for $k \geq 1$

$$G_k = \text{part of } F_k \text{ orthogonal to } G_0, G_1, \ldots, G_{k-1}$$

- Components of statistics

$$\mathcal{Z}_{k,j} = \frac{X_j^T G_k}{\|G_k\|}$$

- Statistics such as $stat_k$ built from $X_j^T(Y - F_{k,-j})$ are linear combinations of these $\mathcal{Z}_{k,j}$

Distribution Evolution

Lemma 1: shifted normal conditional distribution

Given $\mathcal{F}_{k-1} = (\|G_0\|, \ldots, \|G_{k-1}\|, \mathcal{Z}_0, \mathcal{Z}_1, \ldots, \mathcal{Z}_{k-1})$, the $\mathcal{Z}_k$ has the distributional representation

$$\mathcal{Z}_k = \frac{\|G_k\|}{\sigma_k} b_k + Z_k$$

- $\|G_k\|^2/\sigma_k^2 \sim$ Chi-square$(n-k)$
- $b_0, b_1, \ldots, b_k$ the successive orthonormal components of

$$\begin{bmatrix} \beta \\ \sigma \end{bmatrix}, \begin{bmatrix} \hat{\beta}_1 \\ 0 \end{bmatrix}, \ldots, \begin{bmatrix} \hat{\beta}_k \\ 0 \end{bmatrix} \qquad (*)$$

- $Z_k \sim N(0, \Sigma_k)$ indep of $\|G_k\|$
- $\Sigma_k = I - b_0 b_0^T - b_1 b_1^T - \ldots - b_k b_k^T$
  $=$ projection onto space orthogonal to $(*)$
- $\sigma_k^2 = \hat{\beta}_k^T \Sigma_{k-1} \hat{\beta}_k$

Combining Components

Class of statistics $stat_k$ formed by combining $\mathcal{Z}_0, \ldots, \mathcal{Z}_k$

$$stat_{k,j} = \mathcal{Z}_{k,j}^{comb} + \frac{\sqrt{n}}{\sqrt{c_k}} \hat{\beta}_{k,j}$$

with $\mathcal{Z}_k^{comb} = \lambda_{0,k}\, \mathcal{Z}_0 + \lambda_{1,k}\, \mathcal{Z}_1 + \ldots + \lambda_{k,k}\, \mathcal{Z}_k$, $\sum_{k'} \lambda_{k',k}^2 = 1$

Combining Components

Class of statistics $stat_k$ formed by combining $\mathcal{Z}_0, \ldots, \mathcal{Z}_k$

$$stat_{k,j} = \mathcal{Z}_{k,j}^{comb} + \frac{\sqrt{n}}{\sqrt{c_k}}\,\hat{\beta}_{k,j}$$

with $\mathcal{Z}_k^{comb} = \lambda_{0,k}\,\mathcal{Z}_0 + \lambda_{1,k}\,\mathcal{Z}_1 + \ldots + \lambda_{k,k}\,\mathcal{Z}_k$, $\sum_{k'} \lambda_{k',k}^2 = 1$

Ideal Distribution of the combined statistics

$$stat_k^{ideal} = \frac{\sqrt{n}}{\sqrt{c_k}}\beta + Z_k^{comb}$$

for $c_k = \sigma^2 + (1 - x_k)P$

For the terms sent the shift $\alpha_{\ell,k}$ has an effective *snr* interpretation

$$\alpha_{\ell,k} = \sqrt{n\,\frac{P_\ell}{c_k}} = \sqrt{n\,\frac{P_\ell}{\sigma^2 + P_{remaining,k}}}$$

Oracle statistics

Weights of combination: based on $\underline{\lambda}_k$ proportional to

$$\Big((\sigma_Y - b_0^T \hat{\beta}_k),\ -(b_1^T \hat{\beta}_k), \ldots, -(b_k^T \hat{\beta}_k) \Big)$$

Combining $\mathcal{Z}_k$ with these weights, replacing χ_{n-k} with $\sqrt{n}$, it produces the desired distributional representation

$$stat_k = \frac{\sqrt{n}}{\sqrt{\sigma^2 + \|\beta - \hat{\beta}_k\|^2}} \beta + Z_k^{comb}$$

with $Z_k^{comb} \sim N(0, I)$ and $\sigma_Y^2 = \sigma^2 + P$.

Oracle statistics

Weights of combination: based on $\underline{\lambda}_k$ proportional to

$$\Big((\sigma_Y - b_0^T \hat{\beta}_k),\ -(b_1^T \hat{\beta}_k), \ldots, -(b_k^T \hat{\beta}_k) \Big)$$

Combining $\mathcal{Z}_k$ with these weights, replacing χ_{n-k} with $\sqrt{n}$, it produces the desired distributional representation

$$stat_k = \frac{\sqrt{n}}{\sqrt{\sigma^2 + \|\beta - \hat{\beta}_k\|^2}} \beta + Z_k^{comb}$$

with $Z_k^{comb} \sim N(0, I)$ and $\sigma_Y^2 = \sigma^2 + P$.

- Can't calculate the weights not knowing β, e.g. $b_0 = \beta/\sigma$
- $\|\beta - \hat{\beta}_k\|^2$ is close to its known expectation
- Provides the desired distribution of the $stat_{k,j}$

The Ballpark Method of Nearby Measures

- A sequence $\mathbb{P}_L$ of true distributions of the statistics
- A sequence $\mathbb{Q}_L$ of convenient approximate distributions
- $D_\gamma(\mathbb{P}_L \| \mathbb{Q}_L)$, the Renyi divergence between the distributions

$$D_\gamma(\mathbb{P} \| \mathbb{Q}) = (1/\gamma) \log \mathbb{E}[(p(stat)/q(stat))^{\gamma-1}]$$

- A sequence A_L of events of interest
- Lemma: If the Renyi divergence is bounded by a value D, then any event of exponentially small probability using the simplified measures $\mathbb{Q}_L$ also has exponentially small probabity using the true measures $\mathbb{P}_L$

$$\mathbb{P}(A_L) \leq e^{2D}[\mathbb{Q}(A_L)]^{1/2}$$

- With bounded D, allows treating statistics as Gaussian

Approximating distribution for $\mathcal{Z}_k$

We approximate the distribution for $\mathcal{Z}_k$ given $\mathcal{F}_{k-1}$ as

$$\mathcal{Z}_k = \sqrt{n} b_k + Z_k$$

where $Z_k \sim N(0, I - Proj_k)$ where $Proj_k$ is a projection matrix to the space spanned by $(\hat{\beta}_1, \ldots, \hat{\beta}_k)$.

Lemma. For any event A that is determined by the random variables,

$$\|G_{k'}\| \text{ and } \mathcal{Z}_{k'} \text{ for } k' = 0, \ldots, k$$

we have

$$\mathbb{P}A \leq (\mathbb{Q}Ae^{k(2+k^2/n+C)})^{1/2}$$

Statistics based on weights of combination

Oracle weights of combination: $\underline{\lambda}_k$ proportional to

$$\left((\sigma_Y - b_0^T \hat{\beta}_k),\ -(b_1^T \hat{\beta}_k), \ldots, -(b_k^T \hat{\beta}_k) \right)$$

Estimated weights of combination: $\underline{\lambda}_k$ proportional to

$$\left((\|Y\| - \mathcal{Z}_0^T \hat{\beta}_k),\ -(\mathcal{Z}_1^T \hat{\beta}_k), \ldots, -(\mathcal{Z}_k^T \hat{\beta}_k) \right)$$

These estimated weights produce the residual based statistics previously discussed

Orthogonalization Interpretation of Weights

Estimation of

$$\Big((\sigma_Y - b_0^T \hat{\beta}_k),\ -(b_1^T \hat{\beta}_k), \ldots, -(b_k^T \hat{\beta}_k) \Big)$$

These $b_{k'}^T \hat{\beta}_k$ arise in
the QR-decomposition for $B = [\beta, \hat{\beta}_1, \ldots, \hat{\beta}_k]$

$$\mathrm{B} = \begin{bmatrix} b_0 & b_1 & \cdots & b_k \end{bmatrix} \begin{bmatrix} (b_0^T \beta) & (b_0^T \hat{\beta}_1) & \cdots & (b_0^T \hat{\beta}_k) \\ 0 & (b_1^T \hat{\beta}_1) & \cdots & (b_1^T \hat{\beta}_k) \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & (b_k^T \hat{\beta}_k) \end{bmatrix}$$

Cholesky Decomposition for $B^T B$

For given $\hat{\beta}_1, \ldots, \hat{\beta}_k$ and $c_0 > \ldots > c_k$, $\quad c_k = \sigma^2 + (1 - x_k)P$

these $b_{k'}^T \hat{\beta}_k$ arise in the Cholesky decomposition $B^T B = R^T R$,

$$\begin{bmatrix} (\beta^T\beta) & (\beta^T\hat{\beta}_1) & \cdots & (\beta^T\hat{\beta}_k) \\ (\beta^T\hat{\beta}_1) & (\hat{\beta}_1^T\hat{\beta}_1) & \cdots & (\hat{\beta}_1^T\hat{\beta}_k) \\ \vdots & \vdots & \ddots & \vdots \\ (\beta^T\hat{\beta}_k) & \cdots & \cdots & (\hat{\beta}_k^T\hat{\beta}_k) \end{bmatrix} = R^T \begin{bmatrix} (b_0^T\beta) & (b_0^T\hat{\beta}_1) & \cdots & (b_0^T\hat{\beta}_k) \\ 0 & (b_1^T\hat{\beta}_1) & \cdots & (b_1^T\hat{\beta}_k) \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & (b_k^T\hat{\beta}_k) \end{bmatrix}$$

Replace the left side with known deterministic quantities $x_k P$.

Cholesky Decomposition for $B^T B$

For given $\hat{\beta}_1, \ldots, \hat{\beta}_k$ and $c_0 > \ldots > c_k$, $\quad c_k = \sigma^2 + (1 - x_k)P$

these $b_{k'}^T \hat{\beta}_k$ arise in the Cholesky decomposition $B^T B = R^T R$,

$$\begin{bmatrix} \sigma_Y^2 & x_1 P & \cdots & x_k P \\ x_1 P & x_1 P & \cdots & x_1 P \\ \vdots & \vdots & \ddots & \vdots \\ x_k P & \cdots & \cdots & x_k P \end{bmatrix} = R^T \begin{bmatrix} \sigma_Y & (\sigma_Y - c_1\sqrt{\omega_0}) & \cdots & (\sigma_Y - c_k\sqrt{\omega_0}) \\ 0 & c_1\sqrt{\omega_1} & \cdots & c_k\sqrt{\omega_1} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & c_k\sqrt{\omega_k} \end{bmatrix}$$

Replace the left side with known deterministic quantities $x_k P$. Then the right side would be replaced by some deterministic values where $\omega_k = 1/c_k - 1/c_{k-1}$ for $k \geq 1$ and $\omega_0 = 1/c_0$.

This motivates the deterministic weights of combinations

Deterministic weights of combinations

For given $\hat{\beta}_1, \ldots, \hat{\beta}_k$ and $c_0 > \ldots > c_k$, $c_k = \sigma^2 + (1 - x_k)P$
Combine $\mathcal{Z}_{k'} = \sqrt{n}\, b_{k'} + Z_{k'}$ with

$$\underline{\lambda}_k^* = \sqrt{c_k}\, (\sqrt{\omega_0}\,, -\sqrt{\omega_1}\,, \ldots, -\sqrt{\omega_k}\,)$$

$$= \sqrt{c_k} \left(\sqrt{\frac{1}{c_0}}, -\sqrt{\frac{1}{c_1} - \frac{1}{c_0}}, \ldots, -\sqrt{\frac{1}{c_k} - \frac{1}{c_{k-1}}} \right)$$

yielding approximately optimal statistics

$$stat_k = \sum_{k'=0}^{k} \lambda^*_{k',k} \mathcal{Z}_{k'} + \frac{\sqrt{n}}{\sqrt{c_k}} \hat{\beta}_k$$

Reliability under $\mathbb{Q}$

Lemma: For $k = 1, \ldots, k^*$,

$$A_k = \{|\beta^T \hat{\beta}_k / P - x_k| > \eta_k\} \cup \{|\|\hat{\beta}_k\|^2 / P - x_k)| > \eta_k\}$$

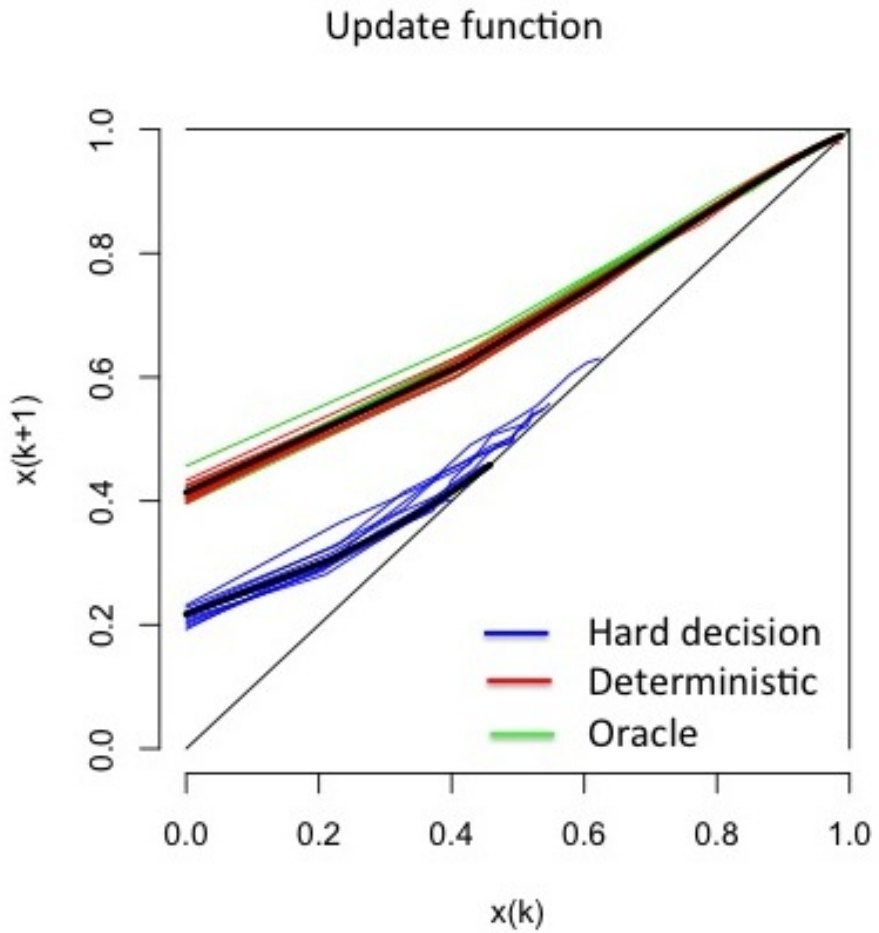
with $\eta_k \sim (n/L)\eta_{k-1}$. Then, we have

$$\mathbb{Q}\{\cup_{k'=1}^{k} A_{k'}\} \lesssim \sum_{k'=1}^{k} 6(k+1) \exp\{-\frac{L}{4c^2}\eta_k^2\}$$

where $c^2 = L \max_\ell (P_\ell / P)$.

Update Plots for Deterministic and Oracle Weights

Figure : $L = 512$, $M = 64$, $snr = 15$, $C = 2$ *bits*, $R = 0.7C$, *blocklength* $= 2194$. Ran 10 experiment for each method. We see that they follow the expected update function.



Cholesky Decomposition-based Estimated Weights

These $b_{k'}^T \hat{\beta}_k$ arise in the Cholesky decomposition $B^T B = R^T R$,

$$\begin{bmatrix} (\beta^T\beta) & (\beta^T\hat{\beta}_1) & \cdots & (\beta^T\hat{\beta}_k) \\ (\beta^T\hat{\beta}_1) & (\hat{\beta}_1^T\hat{\beta}_1) & \cdots & (\hat{\beta}_1^T\hat{\beta}_k) \\ \vdots & \vdots & \ddots & \vdots \\ (\beta^T\hat{\beta}_k) & \cdots & \cdots & (\hat{\beta}_k^T\hat{\beta}_k) \end{bmatrix} = R^T \begin{bmatrix} (b_0^T\beta) & (b_0^T\hat{\beta}_1) & \cdots & (b_0^T\hat{\beta}_k) \\ 0 & (b_1^T\hat{\beta}_1) & \cdots & (b_1^T\hat{\beta}_k) \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & (b_k^T\hat{\beta}_k) \end{bmatrix}$$

Under $\mathbb{Q}$, we have

$$\mathcal{Z}_k^T \hat{\beta}_k / \sqrt{n} = b_k^T \hat{\beta}_k$$

Then, we can recover the rest of the components which leads us to the oracle weights under the approximating distribution and we will denote it by $\hat{\underline{\lambda}}_k$

Cholesky weights of combinations

If we combine $\mathcal{Z}_k$ with weights $\hat{\underline{\lambda}}_k$

$$\hat{stat} = \sum_{k'=0}^{k} \hat{\lambda}_{k,k'} \mathcal{Z}_k + \frac{\sqrt{n}}{\sqrt{\hat{c}_k}} \hat{\beta}_k$$

where $\hat{c}_k = \sigma^2 + \|\beta - \hat{\beta}_k\|^2$, then it produces the desired distributional representation under $\mathbb{Q}$

$$\hat{stat}_k = \frac{\sqrt{n}}{\sqrt{\hat{c}_k}} \beta + Z_k^{comb}$$

Reliability under the $\mathbb{Q}$

Lemma. Suppose we have a Lipschitz condition on the update function with $c_{Lip} \leq 1$ so that

$$|g(x_1) - g(x_2)| \leq c_{Lip}|x_1 - x_2|.$$

For $k = 1, \ldots, k^*$,

$$A_k = \{|\beta^T \hat{\beta}_k / P - x_k| > k\eta\} \cup \{|\|\beta - \hat{\beta}_k\|^2 / P - (1 - x_k)| > k\eta\}$$

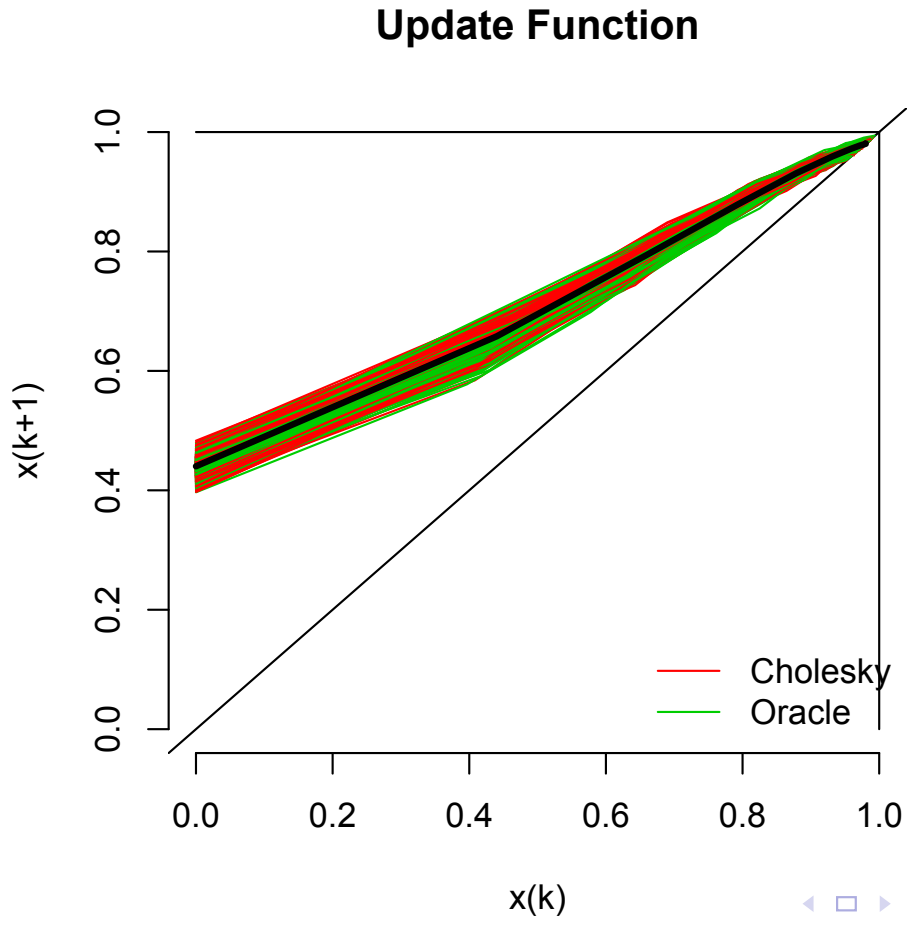
Then, we have

$$\mathbb{Q}\{\cup_{k'=1}^{k} A_{k'}\} \lesssim \exp(-\frac{L}{8c^2}\eta^2)$$

where $c^2 = L \max_\ell(P_\ell / P)$.

Update Plots for Cholesky-based Weights

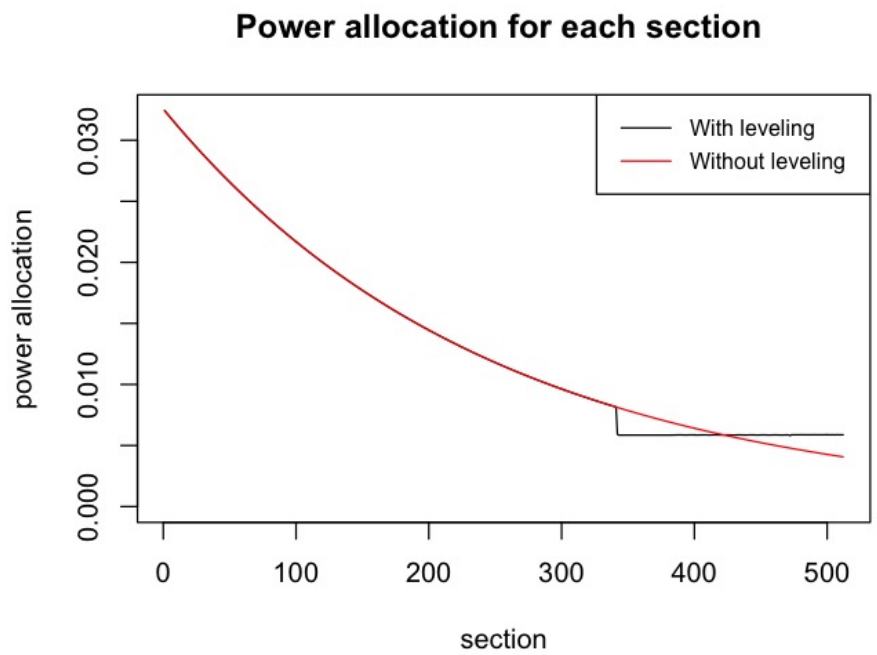
Figure : $L = 512$, $M = 64$, $snr = 7$, $C = 1.5$ *bits*, $R = 0.7C$, *blocklength* $= 2926$. Red (cholesky decomposition based weights); green (oracle weights of combination) . Ran 10 experiment for each.



Improving the End Game

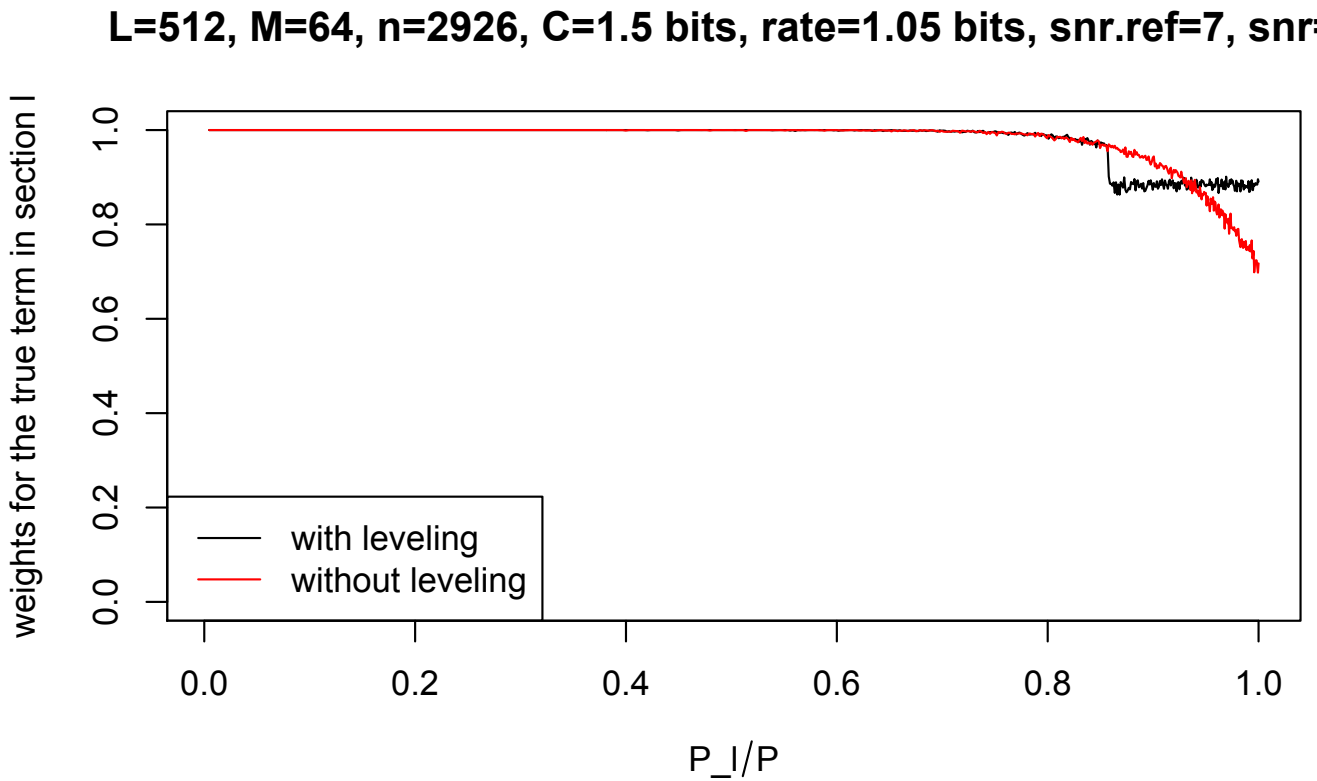
- Variable power: P_ℓ proportional to $e^{-2C\ell/L}$ for $\ell = 1, \ldots, L$
- We use alternative power allocation: constant leveling the power allocation for the last portion of the sections

Figure : $L = 512$, $snr = 7$, $C = 1.5$ *bits*



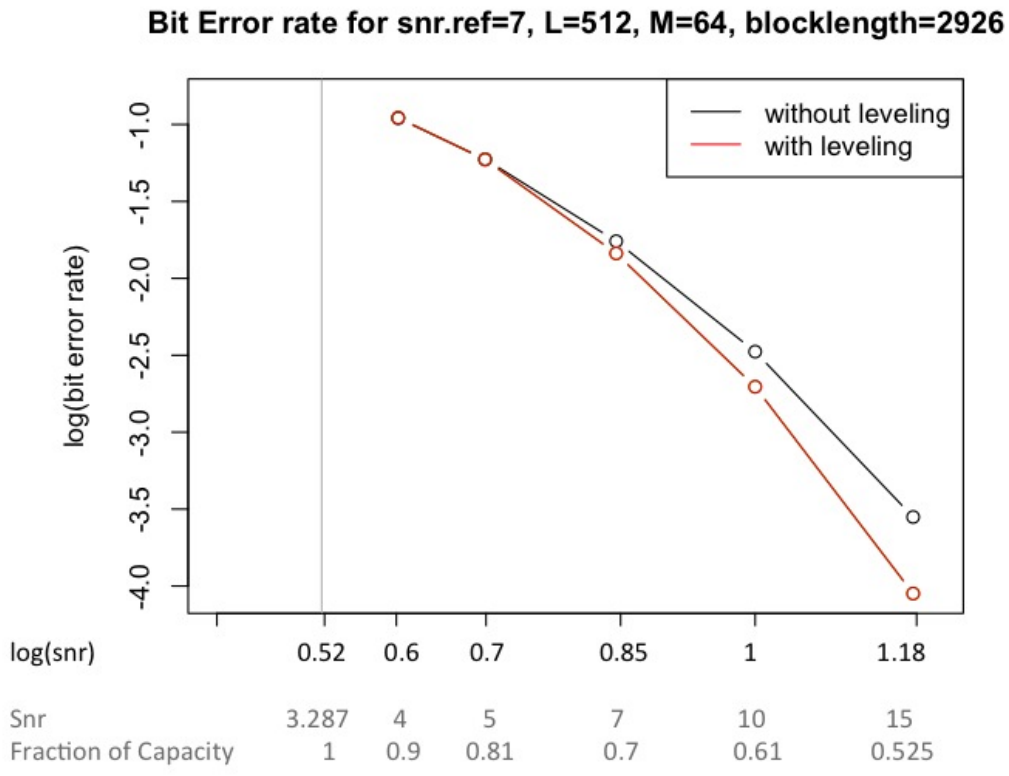
Progression Plot using Alternative Power Allocation

Figure : $L = 512$, $M = 64$, $snr = 7$, $C = 1.5$ *bits*, $R = 0.7C$, *blocklength* $= 2926$. Progression plot of the final step. The area under the curve might be the same, the expected weights for the last sections are higher when we level the power at the end



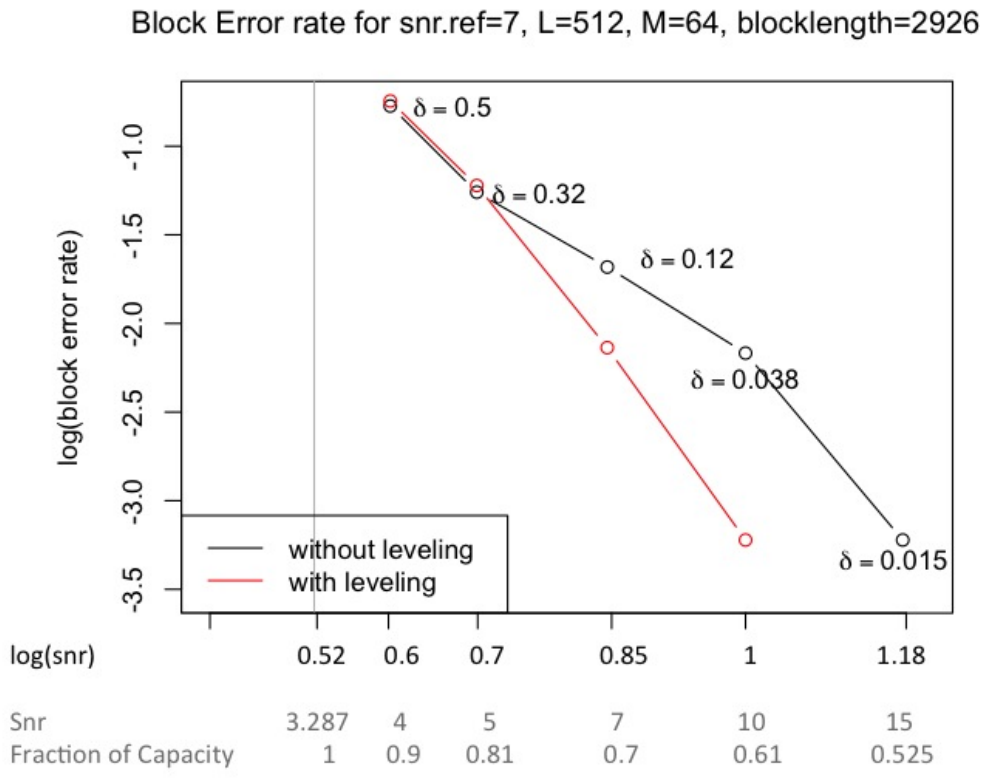
Bit Error Rate

Figure : $L = 512$, $M = 64$, $R = 1.05$ *bits*, *blocklength* $n = 2926$, $snr.ref = 7$, $snr = (4, 5, 7, 10, 15)$. Ran 10,000 trials. Average of error count out of 512 sections.



Block Error Rate

Figure : $L = 512$, $M = 64$, $R = 1.05$ *bits*, *blocklength* $n = 2926$, $snr.ref = 7$, $snr = (4, 5, 7, 10, 15)$. Ran 10,000 trials.



Summary

Sparse superposition codes with adaptive successive decoding

- Simplicity of the code permits:
  - distributional analysis of the decoding progression
  - low complexity decoder
  - exponentially small error probability for any fixed $R < C$
- Asymptotics superior to polar code bounds for such rates

Rate versus Section Size

For Adaptive Successive Decoding with Thresholding

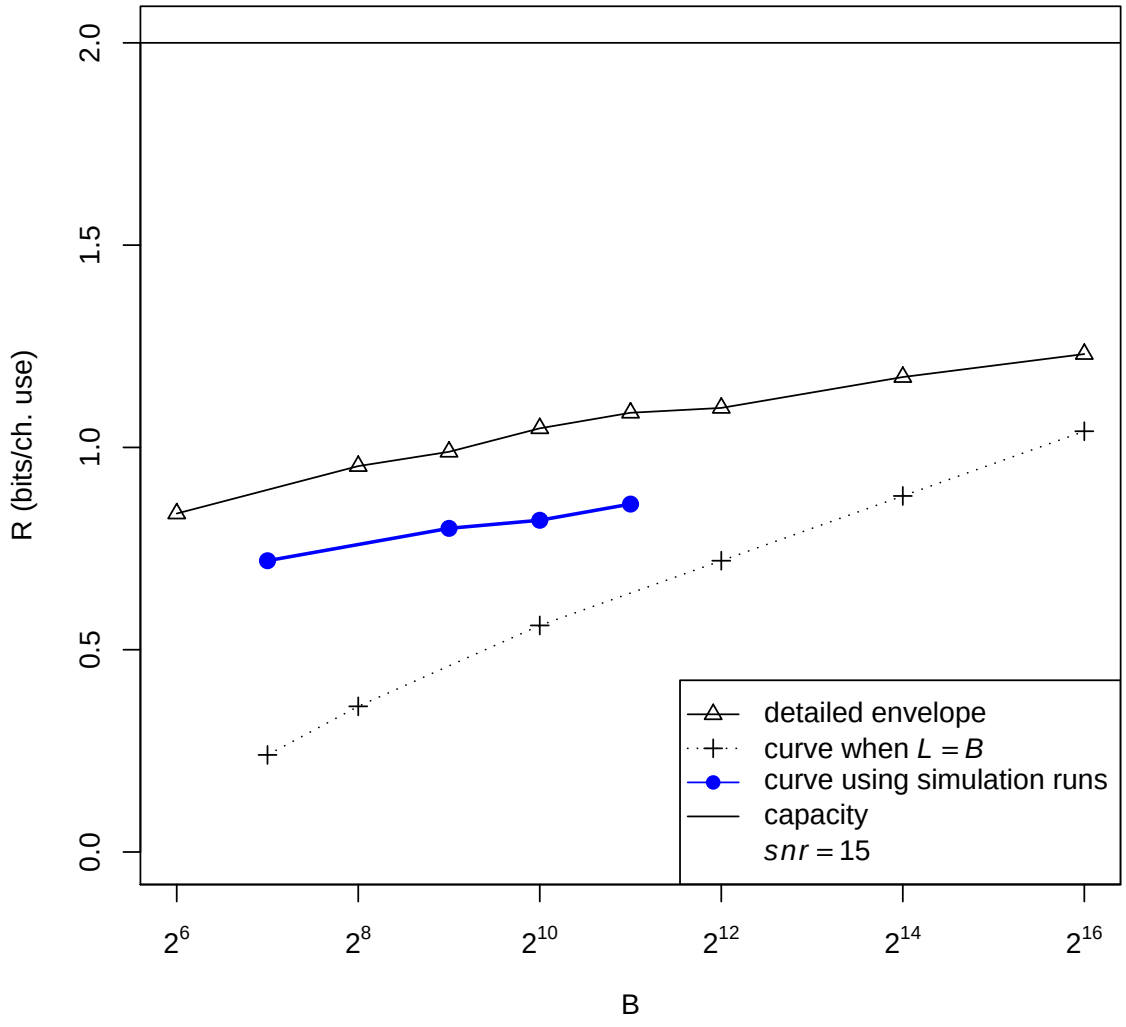


Figure : Rate as function of M for $snr = 15$ and error probability 10^{-3}.

Rate versus Section Size

For Adaptive Successive Decoding with Thresholding

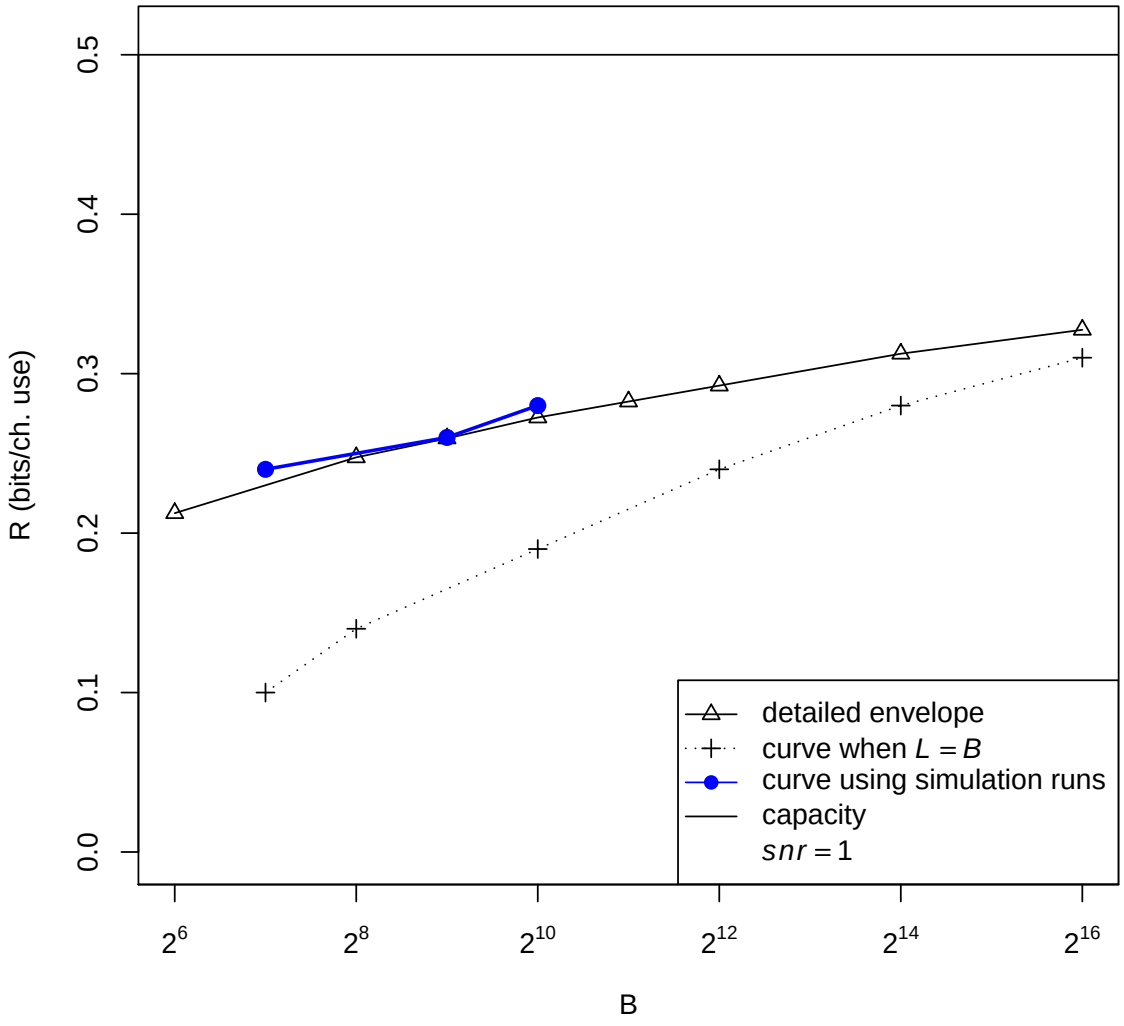


Figure : Rate as function of M for $snr = 1$ and error probability 10^{-3}.