Analysis of Fast Sparse Superposition Codes

Andrew Barron and Antony Joseph
Department of Statistics
Yale University

International Symposium on Information Theory
St. Petersburg, August 4, 2011

Fast Sparse Superposition Codes

Sparse Superposition Codes for the Gaussian Noise Channel

- low complexity
- exponentially small error probability
- for any fixed rate $R < C$

Outline

- Sparse superposition coding
- Adaptive successive decoding
  - simplified form based on residuals
  - refined form based on adaptive orthogonalization
- Distributional analysis
- Rate, reliability, and complexity
  - refined form of exponent
- Comparison with polar codes

Channel Communication Set-up

- 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
- Reliability: Want small Prob$\{\hat{U} \neq U\}$
  and small Prob$\{\textit{Fraction mistakes} \geq \alpha\}$

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

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

$$\downarrow$$

- 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$\{\hat{U} \neq U\}$
  and small Prob$\{\textit{Fraction mistakes} \geq \alpha\}$

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: $\beta = (00*0000000000*00\ldots0*000000)^T$
- 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: small Prob$\{Fraction\ \hat{\beta}\ mistakes \geq \alpha\}$, small α

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 $B = N/L$, 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 $K = L \log \frac{N}{L} = L \log B$

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 $B = N/L$, 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 $K = L \log \frac{N}{L} = L \log B$
- Interpretation: Orthogonal constellations forming β
  are composed with a Gaussian matrix X
  to yield the code vectors

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 $B = N/L$, 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 $K = L \log \frac{N}{L} = L \log B$
- Ultra-sparse case: Impractical $B = 2^{nR/L}$ with L constant
  (reliable at all $R < C$: Cover 1972,1980)
- Moderately-sparse: Practical $B = n$ with $L = nR/\log n$
  (still 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 $B = N/L$, 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 $K = L \log \frac{N}{L} = L \log B$
- Reliability: small Prob$\{\textit{Fraction mistakes} \geq \alpha\}$ 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 $e^{-2C\,\ell/L}$

Adaptive Successive Decoder (simplified version)

Decoding Steps

- 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
  - See which are above threshold
  - Add these columns to the fit
- Stop:
  - At Step $k = 1 + snr \log B$, or
  - if there are no additional inner products above threshold

Complexity of Adaptive Successive Decoder

Complexity in parallel pipelined implementation

- Space: (from $k = snr \log B$ copies of the n by N dictionary)
  - $knN = snr\ CnB$ memory positions
  - kN multiplier/accumulators and comparators
- Time: $O(1)$ per received Y symbol

Rate and Reliability

Result for Optimal ML Decoder, 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 Code

Practical: Adaptive Successive Decoder, with outer RS code.

- prob error exponentially small in $n/(\log B)^{1/2}$ for $R < C$
- Value C_B approaching capacity

$$C_B = \frac{C}{1 + c_1/\log B}$$

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

$$\text{Prob}\{Error\} \le e^{-L(C_B-R)^2 c_2}$$

- Improves to $e^{-c_3 L(C_B-R)^2 (\log B)^{0.5}}$ using a Bernstein bound.
- Nearly optimal when $C_B - R$ is at least $C - C_B$.
- Our c_1 is near $(2.5 + 1/snr)\log\log B + 4C$

Some Relationships to Other Work

- Forney (1960): Concatenated codes
- Barg,Zémor (2002,2004): Expander codes for the BSC:
  - exponential error bounds and linear complexity for $R < C$
- LDPC and turbo codes:
  - some theoretical analysis (Richardson,Urbanke 2008), yet obstacles remain for proof of rates up to capacity
- Arikan polar codes for Gaussian channel (Abbe,Bar.2011):
  - q quantization levels
  - $R < C_q$ with gap $C - C_q \leq snr/q$
  - Error bound from Hassani, Urbanke (2011) in $q = 2$ case:

$$\mathrm{Prob}\{Error\} \leq 2^{-n^{(1-\alpha)/2}(C_q-R)^{1/2\alpha}(1+o(1))}$$

- Tropp 08 codes from compressive sensing; related work:
  - Wainwright; Fletcher, Rangan, Goyal; Zhang; others
  - ℓ_1-constrained least squares practical, has positive rate
  - but not capacity achieving

Adaptive Successive Decoder Ingredients

Intialization: $res_1 = Y$ and $J_1 = \{1, 2, \ldots, N\}$, with $N = LB$

Loop:

- Residual: $res_k = Y - Fit_{k-1}$
- Test Stat: $\mathcal{Z}_{k,j}^{comb} = X_j^T res_k / \|res_k\|$
- Threshold: $\tau = \sqrt{2 \log B} + a$
- Detections: $1_{H_{k,j}} = 1_{\{\mathcal{Z}_{k,j}^{comb} \geq \tau\}}$
- Fit Increment: $F_k = \sum_{j \in J_k} \sqrt{P_j}\, X_j\, 1_{H_{k,j}}$
- Fit Update: $Fit_k = Fit_{k-1} + F_k$
- Remaining: $J_{k+1} = \{j \in J_k : \mathcal{Z}_{k,j}^{comb} < \tau\}$

Distributional analysis of refined decoder

- Test statistic: $\mathcal{Z}_{k,j}^{comb}$
- Test statistic ingredients: $\mathcal{Z}_{k,j} = X_j^{\,T}\, G_k/\|G_k\|$
- The $Y, -F_1, \ldots, -F_{k-1}$ have orthogonalized components

$$G_1, G_2, \ldots, G_k$$

- Approximate distrib: $\mathcal{Z}_{k,j}$ is shift of indep $N(0,1)$ r.v.s

$$\sqrt{(w_k\, u_\ell\, C/R)\, 2\log B}\; 1_{\{j\ sent\}} \;+\; Z_{k,j}$$

with

- $u_\ell = e^{-2C\ell/L}$ for j in section ℓ
- $w_k = s_k - s_{k-1}$ is the increment of the sequence s_k
- s_k equals $s(x) = 1/(1-\nu x) \qquad \nu = snr/(1+snr)$
- evaluated at $x = x_{k-1}$, weighted fraction of prev. detections
- initialized with $x_0 = 0$ and $w_1 = 1$.

Distributional analysis of refined decoder

- Combined test statistic: $\mathcal{Z}_{k,j}^{comb} = \sum_{k'=1}^{k} \sqrt{\lambda_{k'}}\, \mathcal{Z}_{k',j}$
- Best weights $\lambda_{k'} = w_{k'}/s_k$ proportional to w_k
- Approximate distrib: $\mathcal{Z}_{k,j}^{comb}$, maximized shift of $N(0,1)$

$$\sqrt{\frac{(u_\ell\, C/R)\, 2\log B}{1-\nu x}}\; 1_{\{j\ sent\}} \;+\; Z_{k,j}^{comb}$$

  evaluated at $x = x_{k-1}$
- Expected weighted fraction of terms sent above threshold

$$g(x) \;=\; \sum_{\ell=1}^{L} \pi_\ell\, \Phi\left(\tau\left(\sqrt{\frac{u_\ell\, C/R}{1-\nu x}}-1\right)\right)$$

  with $\pi_\ell = P_\ell/P$ proportional to $u_\ell = e^{-2C\ell/L}$
- Provides the performance update $x_k = g(x_{k-1})$
- $g(x)$ shown to exceed x by at least $const/\log B$ for $R \le C_B$

Decoding progression, example bounds

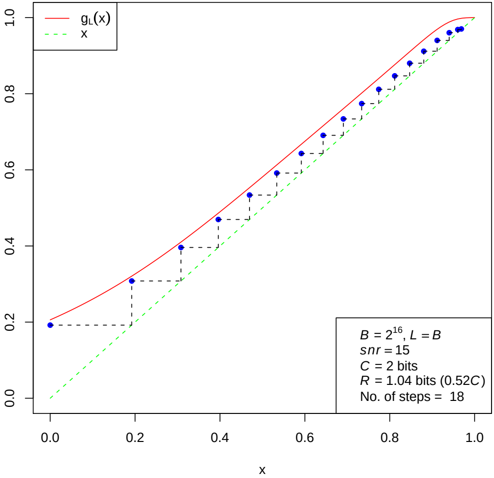

Figure: Plot of $g(x)$ and the sequence x_k for $snr = 15$, with variable power allocation. The threshold uses $a = 0.86$. The final false alarm and failed detection rates are less than 0.026 and 0.013 respectively, with probability of at least that fraction of mistakes less than 0.002.

Decoding Progression

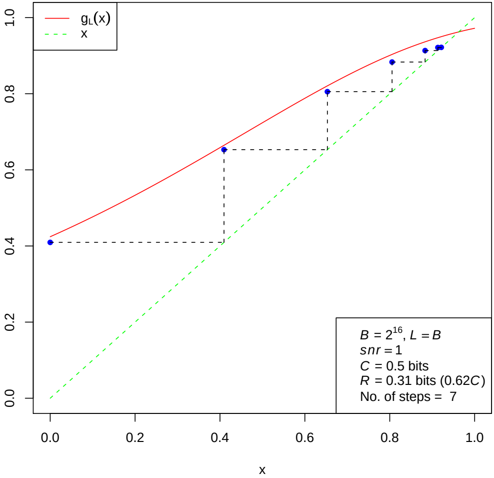

Figure: Plot of $g(x)$ and the sequence x_k for $snr = 1$, with constant power allocation. The threshold uses $a = 0.56$. The final false alarm and failed detection rates are 0.026 and 0.053 respectively, with probability bound 0.0007.

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
- Study of R_n approaching C, e.g. at a polynomial rate (slower than $1/\sqrt{n}$) needs more attention
- Need for new analysis and perhaps new decoding refinements for both types of codes