DIMACS Series in Discrete Mathematics
and Theoretical Computer Science

Interpolation Multiplicity Assignment Algorithms for Algebraic Soft-Decision Decoding of Reed Solomon Codes

Mostafa El-Khamy and Robert J. McEliece

Abstract. In an attempt to determine the ultimate capabilities of the Sudan-Guruswami/Kötter-Vardy algebraic soft-decision decoding algorithm for Reed-Solomon codes, we present a new method, based on the Chernoff bound, for assigning interpolation multiplicities for algebraic soft-decision list decoding. A mathematical framework for optimizing the interpolation multiplicities is laid down. In many cases, the algorithm developed in this paper demonstrates that the potential performance of algebraic soft-decision decoding of Reed-Solomon codes is significantly better than previously thought.

1. Introduction

Reed-Solomon codes [**22**] are one of the most important types of error-correcting codes, due to their wide applicability in data-storage and communication systems. Through the seminal work of Sudan [**23**], Guruswami-Sudan [**8**], and Kötter-Vardy [**11**], we now have a polynomial-time algebraic soft-decision decoding (ASD) algorithm for Reed-Solomon codes. In an attempt to find asymptotic (in decoder complexity) performance limits for ASD, we develop a new class of *multiplicity assignment* algorithms for ASD in this paper. Roughly speaking, the idea is to choose the multiplicity matrix so as to maximize the probability that the causal codeword is on the decoder's list, as suggested by [**18**], rather than to maximize the expected score of the causal codeword, as is done in [**11**]. However, whereas in [**18**], a Gaussian approximation is employed, we use a Chernoff bound instead. (It was independently suggested in [**21**], in a somewhat different context, to use the Chernoff bound in optimizing symbol based multiplicity matrices).

Here is an overview of the paper. Some preliminaries are given in Section 2. In Section 3, a brief overview of the Guruswami-Sudan (GS) algorithm is given. In Sections 4-8, we describe the theory behind our method. A quick review of

2000 *Mathematics Subject Classification.* 94B20, 94B35, 94B70.

Key words and phrases. error correcting codes, Reed-Solomon codes, soft-decision decoding, algebraic soft decoding, list decoding, Guruswami-Sudan algorithm, Chernoff bound, maximum likelihood, burst error correcting codes, interpolation multiplicities.

This research was supported by NSF grant no. CCR-0118670 and grants from Sony, Qualcomm, and the Lee Center for Advanced Networking. The material in this paper was presented in part at the IEEE International Symposium on Information Theory, Chicago, IL, USA in June 2004.

previously proposed multiplicity assignment ASD algorithms is given in section 6. Our algorithm is developed and explained in Sections 9-13. In Section 14, we present some numerical results and discussions. Conclusions and future research directions are offered in Section 15. Briefly, we conclude that our method is theoretically superior to previously proposed ASD algorithms, although whether it will prove to be practical remains to be seen.

2. Preliminaries

Throughout this paper F will denote a finite field with q elements, and a typical element of F will be denoted by β. $\mathbb{C}$ will be an (n, k, d) Reed-Solomon code over F.[1] Let the information data vector of k elements be $\mathbf{m} = (m_0, m_1, \cdots m_{k-1})$. Then the corresponding codeword $\mathbf{c} = (c_1, \ldots, c_n)$ is generated by polynomial evaluation of the information polynomial $\mathcal{M}(x) = \sum_{i=0}^{k-1} m_i x^i$ at n distinct nonzero elements of F constituting the support set of the code, $S = \{s_i; s_i \in F \text{ for } i = 1, 2, \cdots, n\}$. That is $c_i = \mathcal{M}(s_i)$ for $i = 1, 2, ..., n$.

We will often encounter $q \times n$ arrays (or matrices) of real numbers, typically denoted by $W = (w_i(\beta))$, where $i = 1, \ldots, n$ and $\beta \in F$. The *cost* of such an array is defined to be

$$|W| \triangleq \frac{1}{2} \sum_{i=1}^{n} \sum_{\beta \in F} w_i(\beta) \left(w_i(\beta) + 1\right).$$

If $\mathbf{u} = (u_1, \ldots, u_n)$ is a n-dimensional vector over F, the *score* of $\mathbf{u}$ with respect to the array W is is defined to be

$$\langle \mathbf{u}, W \rangle \triangleq \sum_{i=1}^{n} w_i(u_i). \tag{2.1}$$

The underlying (discrete input, memoryless) channel model has input alphabet F, output alphabet R (which could be of infinite size for continuous channels), and transition probabilities $\Pr\{Y = r | X = \beta\}$, where X and Y denote the channel input and output respectively. Given a received symbol $r \in R$, there is a unique *a posteriori* density function on F corresponding to each $\beta \in F$;

$$p_r(\beta) = \Pr\{X = \beta | Y = r\}.$$

Observing a channel output r is therefore equivalent to being given $p_r(\beta)$ for all $\beta \in F$. From this viewpoint, the output alphabet is not R but

$$\mathcal{R} = \{p_r(\beta); r \in R, \beta \in F\}.$$

Thus in this paper we will assume that if $\mathbf{c} = (c_1, \ldots, c_n)$ is transmitted, the received word is an array of density functions $\Pi = (\pi_i(\beta))$, where $\pi_i(\beta) \in \mathcal{R}$, for $i = 1, \ldots, n$ and $\beta \in F$. We call Π the *a posteriori probability*, or APP, matrix. We denote by $\overline{\mathcal{R}}$ the set of all possible APP matrices. It should be noted that the density functions $\pi_i(\beta)$ could be calculated from the soft channel output as is the case for additive white Gaussian noise (AWGN) channels. However, the density functions could also be delivered directly as the soft output of an inner decoder such as the BCJR algorithm [**1**] or the soft output Viterbi algorithm (SOVA) [**9, 24**] in concatenated coding systems.

[1] More precisely, $\mathbb{C}$ may be a coset of the parent RS code. See Section 5.

The indicator function Δ is defined to be

$$\Delta\left[\texttt{condition}\right] = \begin{cases} 1 & \text{if } \texttt{condition} \text{ is true;} \\ 0 & \text{if } \texttt{condition} \text{ is false.} \end{cases} \tag{2.2}$$

Finally, we will denote the ubiquitous quantity $(k-1)$ by v.

3. The Guruswami-Sudan Algorithm.

Given a $q \times n$ array of nonnegative integers $M = (m_i(\beta))$, called a *multiplicity matrix*, the GS algorithm is a list decoding algorithm which produces as an output a list of at most $\sqrt{2|M|/v}$ codewords [**12**], which contains all codewords $\mathbf{c}$ such that

$$\langle \mathbf{c}, M \rangle > D_v(|M|), \tag{3.1}$$

where $D_v(\gamma)$ is the least positive integer D such that $\left|\left\{(i,j) \in \mathbb{N}^2; i + vj \leq D\right\}\right| \geq \gamma + 1$. In other words, $D_v(|M|)$ is the minimal $(1,v)$ weighted degree of a bivariate polynomial $\mathcal{B}(x,y)$ in order for such a nontrivial polynomial that could be interpolated to pass through all the points (s_i, β) with multiplicity at least $m_i(\beta)$ exists. If the sufficient condition of (3.1) is satisfied, then this bivariate polynomial will have a linear factor of the form $y - \mathcal{M}(x)$ where $\mathcal{M}(x)$ has a degree at most v and is the data polynomial associated with the codeword $\mathbf{c}$ [**8, 11**]. Explicitly,

$$D_v(\gamma) = \left\lfloor \frac{\gamma}{m} + \frac{v(m-1)}{2} \right\rfloor, \quad \text{where } m = \left\lfloor \sqrt{\frac{2\gamma}{v} + \frac{1}{4}} + \frac{1}{2} \right\rfloor. \tag{3.2}$$

In the rest of the paper we will denote the important relationship (3.1) by

$$\mathbf{c} \vdash M. \tag{3.3}$$

We conclude this section with two technical results needed later.

LEMMA 3.1. *An upper bound on the function $D_v(\gamma)$ is*

$$D_v(\gamma) \leq -\frac{v}{2} + \sqrt{2v\gamma} + \frac{v^{3/2}}{8\sqrt{2\gamma}}. \tag{3.4}$$

PROOF. Let m be the unique integer satisfying [**16**]

$$\binom{m}{2} \leq \frac{\gamma}{v} < \binom{m+1}{2}. \tag{3.5}$$

Thus, $\gamma \geq \frac{vm(m-1)}{2}$. Let $\psi(m) = \frac{\gamma}{m} + \frac{v(m-1)}{2}$, then $\psi(m) \geq v(m-1)$. Thus,

$$\frac{\partial \psi(m)}{\partial m} \geq v \geq 0,$$

which implies that $\psi(m)$ is a non-decreasing function of m if γ satisfies (3.5). Since $m \leq \left(\sqrt{\frac{2\gamma}{v} + \frac{1}{4}} + \frac{1}{2}\right)$, it follows that

$$D_v(\gamma) = \lfloor \psi(m) \rfloor \leq \psi(m) \leq \psi\left(\sqrt{\frac{2\gamma}{v} + \frac{1}{4}} + \frac{1}{2}\right). \tag{3.6}$$

With some algebra, we get

$$D_v(\gamma) \leq -\frac{v}{2} + \sqrt{2v\gamma + \frac{v^2}{4}} \leq -\frac{v}{2} + \sqrt{2v\gamma}\left(1 + \frac{v}{16\gamma}\right), \tag{3.7}$$

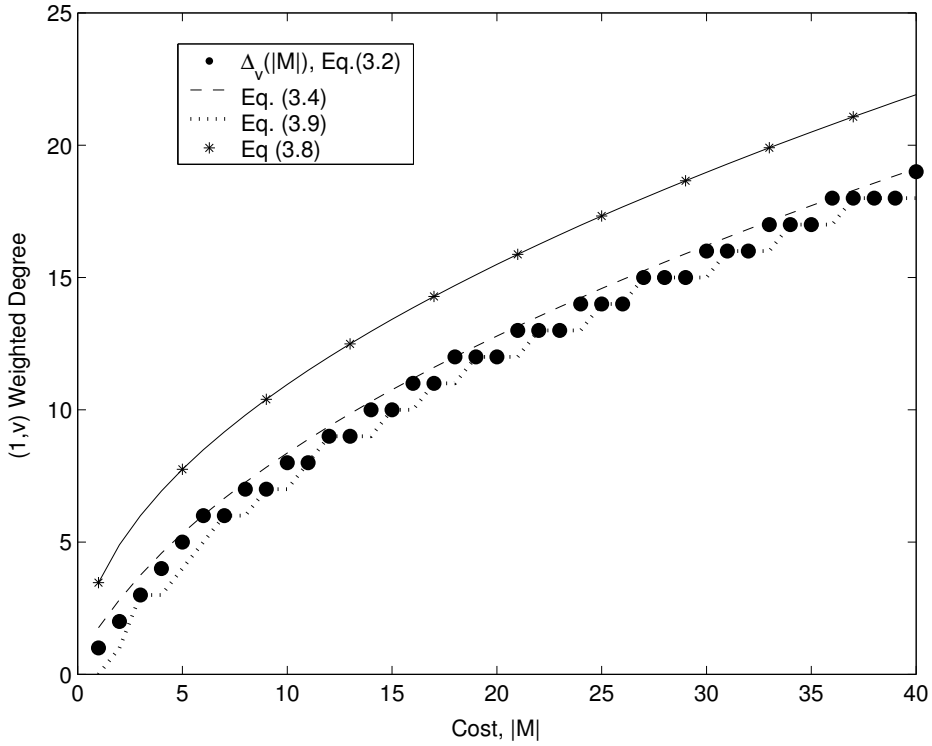

FIGURE 1. Bounds on the function $D_v(|M|)$ as a function of $|M|$ for $v = 6$.

which implies the assertion. □

From the derivation of the above lemma it is clear that

$$D_v(\gamma) \leq \sqrt{2v\gamma} \tag{3.8}$$

is a looser upper bound than that of (3.4). In fact, the function $D_v(\gamma)$ is well approximated by

$$D_v(\gamma) \approx \left\lfloor \sqrt{2v\gamma} - \frac{v}{2} \right\rfloor. \tag{3.9}$$

Indeed, if v is fixed, $0 \leq D_v(\gamma) - \left\lfloor \sqrt{2v\gamma} - \frac{v}{2} \right\rfloor \leq 1$ for all sufficiently large γ. In Figure 1, the discrete function $D_v(|M|)$ is plotted for $v = 6$ as a function of the cost $|M|$. The upper bounds of (3.4) and (3.8) are also plotted. It is clear that the upper bound of (3.4) is a tight (continuous) upper bound. The approximation of (3.9) is also compared to the function $D_v(|M|)$.

LEMMA 3.2. *If* $\gamma > 0$,

$$\lim_{\lambda \to \infty} \frac{D_v(\lambda^2 \gamma)}{\lambda} = \sqrt{2v\gamma}.$$

PROOF. Using (3.4), $\lim_{\lambda\to\infty} \frac{D_v(\lambda^2\gamma)}{\lambda} = \lim_{\lambda\to\infty} \frac{-v}{2\lambda} + \frac{\lambda\sqrt{2v\gamma}}{\lambda} = \sqrt{2v\gamma}$. □

4. A Mathematical Model for ASD Decoding of Reed Solomon Codes.

In this section we describe our model for algebraic soft–decision decoding of RS codes.

A codeword $\mathbf{c} = (c_1, \ldots, c_n)$ which we call the *causal codeword*, is selected at random from $\mathbb{C}$, transmitted over a memoryless channel, and received as the APP matrix $\Pi = (\pi_i(\beta))$ where $i = 1, \ldots, n$ and $\beta \in F$. Given the APP matrix Π, the ASD decoding algorithm converts Π into a $q \times n$ multiplicity matrix M. This multiplicity matrix is forwarded to the GS algorithm, which in turn produces a list of codewords, as described in Section 3. If $\mathbf{c} \vdash M$, then the causal codeword $\mathbf{c}$ will be on the list in which case the decoder is declared to have succeeded.

The situation is summarized by the following chain of random vectors and matrices:[2]

$$\mathbf{c} \to \Pi \xrightarrow{A} M. \tag{4.1}$$

The only quantity in (4.1) under engineering control is the multiplicity algorithm A, so the problem of optimizing the ASD algorithm is equivalent to choosing the right multiplicity algorithm:

$$P(\mathcal{A}) = \min_{A \in \mathcal{A}} \Pr\{\mathcal{E}_A\}, \tag{4.2}$$

where

$$\mathcal{E}_A = \{\mathbf{c} \nvdash M\}, \tag{4.3}$$

and $\mathcal{A}$ is some suitably restricted class of multiplicity algorithms. Note that

$$\Pr\{\mathcal{E}_A\} = \sum_{\Pi \in \overline{\mathcal{R}}} \Pr\{\mathcal{E}_A|\Pi\} \Pr\{\Pi\}, \tag{4.4}$$

so that A minimizes $\Pr\{\mathcal{E}_A\}$ iff it minimizes $\Pr\{\mathcal{E}_A|\Pi\}$ for each APP matrix Π. The following theorem shows that $\Pr\{\mathcal{E}_A|\Pi\}$ depends only on $\mathbb{C}$, Π and M, and so we introduce the notation

$$P_{\mathbb{C}}(\Pi, M) \triangleq \Pr\{\mathcal{E}_A|\Pi\}.$$

THEOREM 4.1. *For* $\mathbf{x} = (x_1, \ldots, x_n) \in F^n$ *define* $\mathbf{P}(\mathbf{x}) = \prod_{i=1}^n \pi_i(x_i)$ *and* $\mathbf{P}(\mathbb{C}) = \sum_{\mathbf{c} \in \mathbb{C}} \mathbf{P}(\mathbf{c})$. *Then*

$$P_{\mathbb{C}}(\Pi, M) = \frac{1}{\mathbf{P}(\mathbb{C})} \sum_{\mathbf{c} \in \mathbb{C}} \Delta[\mathbf{c} \nvdash M] \mathbf{P}(\mathbf{c}). \tag{4.5}$$

PROOF. First,

$$\Pr\{\mathcal{E}_A|\Pi\} = \sum_{\mathbf{c} \in \mathbb{C}} \Delta[\mathbf{c} \nvdash M] \Pr\{\mathbf{c}|\Pi\}.$$

Second (cf. [**11**], Appendix A)

$$\Pr\{\mathbf{c}|\Pi\} = \frac{\mathbf{P}(\mathbf{c})}{\mathbf{P}(\mathbb{C})}.$$

□

[2] In order to minimize our notational complexity, we do not distinguish notationally between a random variable and an instance of the random variable.

5. The Kötter-Vardy Simplification.

In Theorem 4.1, it was implicitly assumed that the channel is memoryless and that the components of $\mathbf{c}$ are uniformly drawn from the field F. But because of the maximal distance separable (MDS) property of RS codes, the elements of any subset of k or fewer components of $\mathbf{c}$ are independent and could be treated as information symbols. However, minimizing $P_{\mathbb{C}}(\Pi, M)$ directly is not easy due to the difficulty of calculating $\mathbf{P}(\mathbb{C})$ for an arbitrary code $\mathbb{C}$ and an arbitrary reliability matrix Π. But the following trick, due essentially to Kötter and Vardy [**11**], allows us to replace the Markov chain (4.1) with

$$\mathbf{x} \to \Pi \xrightarrow{A} M, \tag{5.1}$$

which is identical to (4.1) except that the random codeword drawn uniformly from the code $\mathbf{c} \sim U[\mathbb{C}]$ in (4.1) has been replaced with a random vector $\mathbf{x} \sim U[F^n]$ in (5.1), whose components are independent, where $\mathbf{x} \sim U[\mathcal{X}]$ means that $\mathbf{x}$ is drawn uniformly at random from the space $\mathcal{X}$.

COROLLARY 5.1. *If $\mathbb{C}_1, \ldots, \mathbb{C}_K$ are the cosets of $\mathbb{C}$, with $K = q^{n-k}$, then*

$$\begin{aligned} \sum_{i=1}^{K} \mathbf{P}(\mathbb{C}_i) P_{\mathbb{C}_i}(\Pi, M) &= \sum_{\mathbf{x} \in F^n} \Delta[\mathbf{x} \not\vdash M] \, \mathbf{P}(\mathbf{x}) \\ &\triangleq \mathcal{P}(\Pi, M). \end{aligned} \tag{5.2}$$

Since the left hand-side is an average of the error probability $P_{\mathbb{C}_i}(\Pi, M)$, then $P_{\mathbb{C}_i}(\Pi, M) \le \mathcal{P}(\Pi, M)$ for at least one coset $\mathbb{C}_i$.

6. Review of Multiplicity Assignment Algorithms

Algorithms for assigning interpolation multiplicities for the GS algorithm were proposed based on different criteria [**8, 11, 18, 19, 17**]. We will briefly review two algorithms of particular interest.

The Kötter-Vardy Algorithm: The Kötter-Vardy algorithm finds the multiplicity matrix M that maximizes the expectation of the score, $E\{\langle \mathbf{x}, M \rangle\}$, where $\mathbf{x} \sim U[F^n]$ is an n dimensional random vector of independent components [**11**]. A reduced complexity KV algorithm is [**7**]

$$m_i(\beta) = \lfloor \lambda \pi_i(\beta) \rfloor, \tag{6.1}$$

where $\lambda > 0$ is a complexity parameter determined by $|M|$.

The Gaussian Approximation: By definition (2.1), the score of a random vector with respect to a multiplicity matrix M is a sum of n random variables. Assuming that the n random variables are independent, the distribution of the score is approximated by a Gaussian distribution. Based on this approximation, an iterative algorithm is derived to find the multiplicity matrix of infinite interpolation cost that will minimize the error probability [**18**]. Note however that this approximation is valid if n is sufficiently large.

7. Optimization Problem

In view of Corollary 5.1, in the rest of the paper we will focus on choosing M so as to minimize $\mathcal{P}(\Pi, M)$, with the understanding that upper bounds on $\mathcal{P}(\Pi, M)$ technically apply only to the best cosets of the parent RS code.

Usually the ASD decoder will have a cost restriction, so we introduce the notation

$$P(\Pi,\gamma) = \min_{|M|\le\gamma} \mathcal{P}(\Pi,M) \tag{7.1}$$

$$M(\Pi,\gamma) = \arg_M \min_{|M|\le\gamma} \mathcal{P}(\Pi,M). \tag{7.2}$$

Here $P(\Pi,\gamma)$ is the minimum possible ASD decoder error probability, given Π and an upper bound of γ on the cost of M. The matrix $M(\Pi,\gamma)$ is the optimal multiplicity matrix of cost less than or equal to γ corresponding to the APP matrix Π.

We also define

$$P(\Pi,\infty) \triangleq \lim_{\gamma\to\infty} P(\Pi,\gamma), \tag{7.3}$$

which is the minimum possible decoder error probability, given the APP matrix Π, without regard to cost.

Finally, let us consider (cf. (4.2)) the problem of computing

$$P(\gamma) \triangleq \min_{|M|\le\gamma} \Pr\{\mathcal{E}_A\}, \tag{7.4}$$

the minimum possible ASD decoder error probability for decoder cost $\le \gamma$, and

$$P(\infty) \triangleq \lim_{\gamma\to\infty} P(\gamma), \tag{7.5}$$

the absolute minimum ASD decoder error probability, regardless of cost. By (4.4) we have

$$P(\gamma) = \sum_{\Pi\in\overline{\mathcal{R}}} P(\Pi,\gamma)\Pr\{\Pi\} \tag{7.6}$$

$$P(\infty) = \sum_{\Pi\in\overline{\mathcal{R}}} P(\Pi,\infty)\Pr\{\Pi\}. \tag{7.7}$$

8. Soft Multiplicity Matrices

It is difficult to deal with the requirement that the entries of M are integers, so we now define a slightly different problem in which the integer constraint is relaxed and the multiplicities can be arbitrary (nonnegative) real numbers.

Thus let $Q = (q_i(\beta))$ be a "soft" multiplicity matrix, i.e., for each $i = 1,\dots,n$, and each $\beta \in F$, $q_i(\beta)$ is a nonnegative real number. We define

$$\mathcal{P}(\Pi,Q) \triangleq \sum_{\mathbf{x}\in F^n} \Delta[\mathbf{x} \not\vdash Q]\,\mathbf{P}(\mathbf{x}) \tag{8.1}$$

$$P^*(\Pi,\gamma) \triangleq \min_{|Q|\le\gamma} \mathcal{P}(\Pi,Q) \tag{8.2}$$

$$Q^*(\Pi,\gamma) \triangleq \arg\min_{|Q|\le\gamma} \mathcal{P}(\Pi,Q) \tag{8.3}$$

$$P^*(\Pi,\infty) \triangleq \lim_{\gamma\to\infty} P^*(\Pi,\gamma). \tag{8.4}$$

These quantities are the same as the corresponding unstarred ones, (7.1), (7.2), and (7.3), except that the integral matrices (with integer elements) M are replaced

with real matrices Q, so that logically

$$P^*(\Pi,\gamma) \leq P(\Pi,\gamma) \tag{8.5}$$

$$P^*(\Pi,\infty) \leq P(\Pi,\infty). \tag{8.6}$$

Surprisingly, if cost is no object, we loose nothing by relaxing the constraint that the multiplicities be integers. In the following lemma, we show that up-scaling a multiplicity matrix Q with a scalar $\lambda > 1$, results in a lower error probability at the expense of a larger interpolation cost.

LEMMA 8.1. *For any* (Π, Q),

$$\lim_{\lambda\to\infty} \mathcal{P}(\Pi,\lambda Q) \leq \mathcal{P}(\Pi,Q). \tag{8.7}$$

PROOF. Suppose $\Delta[\mathbf{x} \vdash Q] = 1$, then with high probability this implies that

$$\langle \mathbf{x}, Q\rangle \geq \sqrt{2v|Q|} \tag{8.8}$$

for reasonably large costs $|Q|$. If $\lambda \geq 1$, $|\lambda Q| \leq \lambda^2|Q|$, and

$$\frac{\langle \mathbf{x}, \lambda Q\rangle}{D_v(|\lambda Q|)} \geq \frac{\lambda\langle \mathbf{x}, Q\rangle}{D_v(\lambda^2|Q|)}. \tag{8.9}$$

But by Lemma 3.2, the limit of the RHS of (8.9) is $\langle \mathbf{x}, Q\rangle/\sqrt{2v|Q|} \geq 1$, with high probability, where the last inequality follows from (8.8). Thus $\lim_{\lambda\to\infty}\Delta[\mathbf{x} \vdash \lambda Q] = 1$. It follows that for any $\mathbf{x}$,

$$\lim_{\lambda\to\infty}\left\{\sum_{\mathbf{x}\in F^n}\Delta[\mathbf{x} \nvdash \lambda Q]\,\mathbf{P}(\mathbf{x})\right\} \leq \sum_{\mathbf{x}\in F^n}\Delta[\mathbf{x} \nvdash Q]\,\mathbf{P}(\mathbf{x}). \tag{8.10}$$

Comparing this to (8.1), we're done. □

THEOREM 8.2. $P^*(\Pi,\infty) = P(\Pi,\infty)$

PROOF. Define P^+ to denote rational matrices. Then

$$P^*(\Pi,\infty) = P^+(\Pi,\infty), \tag{8.11}$$

by continuity, and

$$P^+(\Pi,\infty) = P(\Pi,\infty), \tag{8.12}$$

by the following argument. If Q is rational, then λQ is integral for arbitrarily large values of λ. Then lemma 8.1 and (8.6) imply (8.12). □

9. The Chernoff Bound.—Finite Cost

We have seen that the number $P^*(\Pi,\gamma)$ (see (8.2), above), delimits the best possible ASD decoding performance, if the APP matrix Π is given. Unfortunately, however, it is very difficult to compute $P^*(\Pi,\gamma)$. In this section, we derive a Chernoff bound on $P^*(\Pi,\gamma)$ (see (9.9), below), which is easy to compute.

Let $\Omega = (F^n, \Pi)$ be a discrete sample space, i.e., for $\mathbf{x} = (x_1,\ldots,x_n) \in F^n$ and $\Pi = (\pi_i(\beta))$ define the probability measure $\mathbf{P}(\mathbf{x}) = \prod_{i=1}^n \pi_i(x_i)$. Define (independent) random variables $\mathcal{S}_1,\ldots,\mathcal{S}_n$ by

$$\mathcal{S}_i(\mathbf{x}) = q_i(x_i) \quad \text{for } i = 1,\ldots,n. \tag{9.1}$$

where $Q = ((q_i(\beta))$ is the multiplicity matrix, and the score

$$S_Q = \langle \mathbf{x}, Q\rangle = \mathcal{S}_1 + \cdots + \mathcal{S}_n. \tag{9.2}$$

Now we have

$$\Pr\{S_Q \le \delta\} = \sum_{\mathbf{x}\in F^n} \Delta\left[\langle \mathbf{x}, Q\rangle \le \delta\right] \mathbf{P}(\mathbf{x}). \tag{9.3}$$

Let $\phi_i(s, \pi_i, q_i)$ be the moment generating function for $\mathcal{S}_i$, i.e.,

$$\phi_i(s, \pi_i, q_i) = E_{\mathcal{S}_i}\left\{e^{s\mathcal{S}_i}\right\} = \sum_{\beta\in F} \pi_i(\beta) e^{s q_i(\beta)}. \tag{9.4}$$

Then the moment generating function for S_Q is

$$\Phi(s, \Pi, Q) = \sum_t \Pr\{S_Q = t\} e^{st} = E_{S_Q}\left\{e^{sS_Q}\right\} \tag{9.5}$$

$$= E_{S_Q}\left\{e^{s\sum_{i=1}^n \mathcal{S}_i}\right\} = E_{S_Q}\left\{\prod_{i=1}^n e^{s\mathcal{S}_i}\right\} \tag{9.6}$$

$$= \prod_{i=1}^n E_{\mathcal{S}_i}\left\{e^{s\mathcal{S}_i}\right\} = \prod_{i=1}^n \phi_i(s, \pi_i, q_i), \tag{9.7}$$

where the expectation and the product are interchanged due to the assumption that the random variables $\mathcal{S}_i$ are independent (see Section 5). Then by the the Chernoff bound (cf. [**20, 25**]),

$$\begin{aligned}\Pr\{S_Q \le \delta\} &= \sum_{t\le\delta} \Pr\{S_Q = t\} \\ &\le \min_{s\ge 0}\left\{\sum_t \Pr\{S_Q = t\} e^{s(\delta - t)}\right\} = \min_{s\ge 0}\left\{e^{s\delta}\Phi(-s, \Pi, Q)\right\}.\end{aligned} \tag{9.8}$$

Finally, if we recall that $P^*(\Pi, \gamma) \triangleq \min_{|Q|\le\gamma} \mathcal{P}(\Pi, Q)$ we have

$$P^*(\Pi, \gamma) \le P^\chi(\Pi, \gamma) \triangleq \min_{|Q|=\gamma, s\ge 0}\left\{e^{sD_v(\gamma)}\Phi(-s, \Pi, Q)\right\}. \tag{9.9}$$

It is a bit awkward to deal with the constraint $|Q| = \gamma$ in (9.9). We could replace this constraint with the more natural constraint $\|\mathbb{X}\|^2 = \sum_{i,\beta} X_i(\beta)^2 = L^2$, where $\mathbb{X} = (X_i(\beta))$ is of the same size as Q, by the following transformation:

$$X_i(\beta) = q_i(\beta) + 1/2; \quad L^2 = 2\gamma + \frac{nq}{4}; \quad D' = D_v(\gamma) + \frac{n}{2}. \tag{9.10}$$

Thus (9.9) could be written as

$$P^*(\Pi, \gamma) \le \min_{\|\mathbb{X}\|^2 = L^2} \min_{s\ge 0}\left\{e^{sD'}\Phi(-s, \Pi, \mathbb{X})\right\}, \tag{9.11}$$

and the optimum matrix is given by

$$\mathbb{X}^* = \arg_{\mathbb{X}} \min_{\|\mathbb{X}\|^2 = L^2} \min_{s\ge 0}\left\{e^{sD'}\Phi(-s, \Pi, \mathbb{X})\right\}. \tag{9.12}$$

10. The Chernoff Bound—Infinite Cost.

In this section, we derive a methodology for performance analysis at asymptotically large costs. We begin by defining an auxiliary function $G^*(\Pi, \zeta)$:

$$G^*(\Pi, \zeta) = \min_{\|R\|^2=1} \sum_{\mathbf{x}\in F^n} \Delta\left[\langle \mathbf{x}, R\rangle \le \zeta\right] \mathbf{P}(\mathbf{x}). \tag{10.1}$$

In the following theorem, we shall see that the case of $\gamma \to \infty$ is the special case of $L^2 = 1$ and $D' = \sqrt{v}$.

THEOREM 10.1. $P^*(\Pi, \infty) = \lim_{\gamma\to\infty} P^*(\Pi, \gamma) = G^*(\Pi, \sqrt{v})$.

PROOF. Define $R = \mathbb{X}/\|\mathbb{X}\|$, then $\|R\|^2 = 1$. By using (9.10) and Lemma 3.1,

$$\lim_{\gamma\to\infty} \frac{D'}{L} = \lim_{\gamma\to\infty} \frac{\sqrt{v} + \frac{v^{3/2}}{16\gamma} + \frac{n-v}{2\sqrt{2\gamma}}}{\sqrt{1 + \frac{nq}{8\gamma}}}. \tag{10.2}$$

Specifically, for large γ the R.H.S. of (10.2) is approximated by

$$\sqrt{v} + \frac{v^{3/2}}{16\gamma} + \frac{n-v}{2\sqrt{2\gamma}} + \left(\sqrt{v} + \frac{v^{3/2}}{16\gamma} + \frac{n-v}{2\sqrt{2\gamma}}\right)\left(-\frac{1}{2}\frac{nq}{8\gamma} + \frac{1.3}{2.4}\left(\frac{nq}{8\gamma}\right)^2 + \dots\right) \to \sqrt{v} + o(1),$$

where $o(1) \to 0$ as $\gamma \to \infty$. Thus,

$$\lim_{\gamma\to\infty} \min_{\|\mathbb{X}\|^2=L^2} \Pr\left\{S_{\mathbb{X}} \le D'\right\} = \lim_{\gamma\to\infty} \min_{\|R\|^2=1} \Pr\left\{S_R \le D'/L\right\} = \min_{\|R\|^2=1} \Pr\left\{S_R \le \sqrt{v}\right\}$$

which by comparing with (9.11) implies the assertion. □

COROLLARY 10.2. $P(\Pi, \infty) = P^*(\Pi, \infty) = G^*(\Pi, \sqrt{v})$.

PROOF. By Theorem 8.2 and Theorem 10.1 we are done. □

Thus $G^*(\Pi, \sqrt{k-1})$ is the minimum possible decoder error probability for the ASD decoder, given the APP matrix Π. Similarly,

$$P(\infty) = \sum_{\Pi\in\overline{\mathcal{R}}} G^*(\Pi, \sqrt{k-1}) \Pr\left\{\Pi\right\}, \tag{10.3}$$

is the unconditional minimum possible decoder error probability. The quantity $G^*(\Pi, \sqrt{v})$, like its finite-cost counterpart $P^*(\Pi, \gamma)$, is difficult to compute exactly, but easy to approximate with the Chernoff bound. To summarize: suppose $R = (r_i(\beta))$, with $\|R\|^2 = 1$ is given. On the $\Omega = \{F^n, \Pi\}$ sample space, define corresponding random variables $\mathcal{R}_i = r_i(x_i)$, for $i = 1, \dots, n$. Then

$$G^*(\Pi, \zeta) = \min_{\|R\|^2=1} \Pr\left\{\mathcal{R}_1 + \dots + \mathcal{R}_n \le \zeta\right\}. \tag{10.4}$$

Let

$$\gamma_i(s, \pi_i, r_i) = \sum_{x\in F} \pi_i(x) e^{s r_i(x)} \tag{10.5}$$

be the moment generating function for $\mathcal{R}_i$, $i = 1, \dots, n$. Then the moment generating function for $S_R = \mathcal{R}_1 + \dots + \mathcal{R}_n$ is

$$\Gamma(s, \Pi, R) = \prod_{i=1}^{n} \gamma_i(s, \pi_i, r_i), \tag{10.6}$$

and the Chernoff bound says that

$$\Pr\{S_n \le \zeta\} \le \min_{s\ge 0}\left\{\Gamma(-s,\Pi,R)e^{s\zeta}\right\}. \tag{10.7}$$

Thus if we define

$$G^{\chi}(\Pi,\zeta) = \min_{\|R\|^2=1}\min_{s\ge 0}\left\{\Gamma(-s,\Pi,R)e^{s\zeta}\right\} \quad \text{and} \tag{10.8}$$

$$R^{\chi}(\Pi,\zeta) = \arg_R \min_{\|R\|^2=1}\min_{s\ge 0}\left\{\Gamma(-s,\Pi,R)e^{s\zeta}\right\}, \tag{10.9}$$

we have the following theorem.

THEOREM 10.3. $P(\Pi,\infty) = P^*(\Pi,\infty) = G^*(\Pi,\sqrt{v}) \le G^{\chi}(\Pi,\sqrt{v})$.

The function $G^{\chi}(\Pi,\sqrt{v}) = G^{\chi}(\Pi,\sqrt{k-1})$ is our main tool, since it is (a) relatively easy to calculate, and (b) a tight upper bound on $P(\Pi,\infty)$, at least when $P(\Pi,\infty)$ is small. Furthermore, the matrix $R^{\chi}(\Pi,\sqrt{k-1})$, when appropriately scaled and quantized, represents a near-optimal choice for the multiplicity matrix for large values of the cost. In the next section, we derive key equations which form the heart of the algorithm used to find the near-optimum multiplicity matrices.

11. The Lagrangian

In this section, we will focus on finding the optimum matrix $\mathbb{X} = (X_i(\beta))$ with a finite cost γ and with L^2 and D' defined as in (9.10). As seen in the previous section, the case of an optimum infinite-cost multiplicity matrix is the special case with $L^2 = 1$ and $D' = \sqrt{v}$. The problem of finding the optimum matrix, $\mathbb{X}^*$, in (9.12) could be reformulated as the constrained optimization problem,

$$\min\left(sD' + \sum_{i=1}^{n}\ln\phi_i(-s,\pi_i,X_i)\right) \tag{11.1}$$

subject to

$$s \ge 0$$
$$\|\mathbb{X}\|^2 = L^2 = 2\gamma + \frac{1}{4}nq.$$

Define the Lagrangian,

$$\mathcal{L}(s,\mathbb{X},\lambda) = sD' + \sum_{i=1}^{n}\ln\phi_i(-s,\pi_i,X_i) + \frac{\lambda}{2}\left(\|\mathbb{X}\|^2 - L^2\right).$$

It is required to solve for $s^*, \mathbb{X}^*$, and λ^* that satisfy

$$\left.\frac{\partial\mathcal{L}}{\partial\lambda}\right|_{\lambda=\lambda^*} = 0, \quad \left.\frac{\partial\mathcal{L}}{\partial s}\right|_{s=s^*} = 0 \quad \text{and} \quad \left.\frac{\partial\mathcal{L}}{\partial X_i(\beta)}\right|_{\mathbb{X}=\mathbb{X}^*} = 0.$$

If the optimization for s results in a negative value for s^*, then this value is discarded and s^* is taken to be at the boundary, i.e. $s^* = 0$. (This may be the case at low signal to noise ratios when the matrix Π has a random like structure.) The corresponding optimized multiplicity matrix X^* is calculated by optimizing for X.

Since $D' = D_v(\gamma) + n/2$ and $\gamma = (\|\mathbb{X}\|^2 - \frac{nq}{4})/2$, then D' is a function of $\mathbb{X}$. Since $D_v(\gamma)$ is actually a discrete function, then it could not be differentiated, however it is well approximated by the continuous upper bound in (3.4),

$$\frac{\partial D'}{\partial X_i(\beta)} \approx \left(\frac{\sqrt{v}}{\sqrt{\|\mathbb{X}\|^2 - \frac{nq}{4}}} - \frac{v^{3/2}}{8\left(\|\mathbb{X}\|^2 - \frac{nq}{4}\right)^{3/2}} \right) X_i(\beta) = \psi(\|\mathbb{X}\|^2) X_i(\beta).$$

In fact the term $\psi(\|\mathbb{X}\|^2)$ will cancel while solving for $\mathbb{X}^*$ below. Solving for $\mathbb{X}^*$ and s^*;

$$\left.\frac{\partial \mathcal{L}}{\partial \lambda}\right|_{\lambda=\lambda^*} = 0 \;\Rightarrow\; \|\mathbb{X}\|^2 = L^2, \tag{11.2}$$

$$\left.\frac{\partial \mathcal{L}}{\partial s}\right|_{s=s^*} = D' - \sum_{i=1}^{n} \left(\frac{\sum_{\beta \in F} X_i(\beta)\pi_i(\beta) e^{-sX_i(\beta)}}{\phi_i(-s, \pi_i, X_i)} \right) \Bigg|_{s=s^*} = 0, \tag{11.3}$$

$$\left.\frac{\partial \mathcal{L}}{\partial X_i(\beta)}\right|_{\mathbb{X}=\mathbb{X}^*} = s\psi(\|\mathbb{X}\|^2) X_i(\beta) - s\frac{\pi_i(\beta) e^{-sX_i(\beta)}}{\phi_i(-s, \pi_i, X_i)} + \lambda X_i(\beta) \Big|_{\mathbb{X}=\mathbb{X}^*} = 0. \tag{11.4}$$

Multiplying (11.4) by $X_i(\beta)$, summing over $\beta \in F$ and then summing over i, we get

$$s\psi(\|\mathbb{X}\|^2)\|\mathbb{X}\|^2 - s\sum_{i=1}^{n} \left(\frac{\sum_{\beta \in F} X_i(\beta)\pi_i(\beta) e^{-sX_i(\beta)}}{\phi_i(-s, \pi_i, X_i)} \right) + \lambda \|\mathbb{X}\|^2 \Big|_{\mathbb{X}=\mathbb{X}^*} = 0. \tag{11.5}$$

Substituting (11.2) and rearranging;

$$\lambda = s\left(\frac{1}{L^2} \sum_{i=1}^{n} \left(\frac{\sum_{\beta \in F} X_i(\beta)\pi_i(\beta) e^{-sX_i(\beta)}}{\phi_i(-s, \pi_i, X_i)} \right) - \psi(L^2) \right). \tag{11.6}$$

Substituting back in (11.4) we reach the following equation

$$\frac{X_i(\beta)}{L^2} \sum_{i=1}^{n} \left(\frac{\sum_{\beta \in F} X_i(\beta)\pi_i(\beta) e^{-sX_i(\beta)}}{\phi_i(-s, \pi_i, X_i)} \right) - \frac{\pi_i(\beta) e^{-sX_i(\beta)}}{\phi_i(-s, \pi_i, X_i)} \Big|_{\mathbb{X}=\mathbb{X}^*} = 0. \tag{11.7}$$

If $s = s^*$, then this equation reduces to

$$\frac{D'}{L^2} X_i(\beta) - \frac{\pi_i(\beta) e^{-s^* X_i(\beta)}}{\sum_{\beta \in F} \pi_i(\beta) e^{-s^* X_i(\beta)}} \Big|_{\mathbb{X}=\mathbb{X}^*} = 0. \tag{11.8}$$

In summary, the optimization problem is reduced to finding s^* and $\mathbb{X}^*$ which are the solutions for equations (11.3) and (11.7) (or (11.8)), respectively.

12. Convexity

In this section, we show that the optimized Lagrangian, $\mathcal{L}^*(s, \mathbb{X}) = \mathcal{L}(s, \mathbb{X}, \lambda^*)$, is convex in both s and $\mathbb{X}$. Thus an iterative algorithm that will minimize $\mathcal{L}^*(s, \mathbb{X})$ could be developed. Specifically we show that for a given multiplicity matrix X', the optimized Lagrangian is convex in the parameter s, and for a given s (at $s = s*$), the optimized Lagrangian is convex in the nq variables which are the components of the multiplicity matrix $\mathbb{X}$. Let

$$\mathcal{L}_s(s) \triangleq \mathcal{L}^*(s, \mathbb{X})|_{\mathbb{X}=X'} \tag{12.1}$$

$$\mathcal{L}_{\mathbb{X}}(\mathbb{X}) \triangleq \mathcal{L}^*(s, \mathbb{X})|_{s=s^*}. \tag{12.2}$$

12.1. $\mathcal{L}_s(s)$ is convex in s.

The gradient of $\mathcal{L}_s(s)$ is defined to be $G_s(s) = \frac{\partial \mathcal{L}_s(s)}{\partial s}$ and is given by (11.3). The second derivative of $\mathcal{L}_s(s)$ with respect to s is

$$\frac{\partial^2 \mathcal{L}_s(s)}{\partial s^2} = \sum_{i=1}^{n} \left(\frac{\sum_{\beta \in F} X_i^2(\beta) \pi_i(\beta) e^{-sX_i(\beta)}}{\sum_{\beta \in F} \pi_i(\beta) e^{-sX_i(\beta)}} - \left(\frac{\sum_{\beta \in F} X_i(\beta) \pi_i(\beta) e^{-sX_i(\beta)}}{\sum_{\beta \in F} \pi_i(\beta) e^{-sX_i(\beta)}} \right)^2 \right).$$

Define the $q \times 1$ dimensional vectors Λ_i and Θ_i such that

$$\Lambda_i = \left\{ X_i(\beta) \sqrt{\pi_i(\beta)} e^{-sX_i(\beta)/2} \right\} \text{ and } \Theta_i = \left\{ \sqrt{\pi_i(\beta)} e^{-sX_i(\beta)/2} \right\} \text{ for } \beta \in F,$$

then the second derivative of $\mathcal{L}_s(s)$ with respect to s is reformulated as

$$H_s = \frac{\partial^2 \mathcal{L}_s(s)}{\partial s^2} = \sum_{i=1}^{n} \left(\frac{\|\Lambda_i\|^2 \|\Theta_i\|^2 - (\Lambda_i^T \Theta_i)^2}{\|\Theta_i\|^4} \right),$$

where for any vector $\mathbf{x}$, $\|\mathbf{x}\| = \left(\mathbf{x}^T \mathbf{x}\right)^{1/2}$ is the Euclidean norm of $\mathbf{x}$. By the Cauchy Schwartz inequality

$$\|\Lambda_i\| \|\Theta_i\| \geq (\|\Lambda_i^T \Theta_i)\|_1,$$

where $\|.\|_1$ is absolute value and $(.)^T$ is the vector transposed, with equality if and only if there exists an $\alpha \geq 0$ such that $\Lambda_i = \alpha \Theta_i$. Thus $H_s \geq 0$, which implies that $\mathcal{L}_s(s)$ is *convex*. In fact, $H_s = 0$ if and only if for each $i = 1, ..., n$, $X_i(\beta) = \alpha_i$ where $\alpha_i \geq 0$ for all $\beta \in F$. Since $X_i(\beta)$ is a function of $\pi_i(\beta)$, then this implies that for each i, $\pi_i(\beta) = 1/q$. This would imply that all symbols $\beta \in F$ are equally likely given the received symbol. At reasonable operating conditions, such a condition does not occur for all $i = 1, .., n$, as it is equivalent to receiving all n symbols of the codeword in error. So in general, $H_s > 0$ and $\mathcal{L}_s(s)$ is *strongly convex* in s.

12.2. $\mathcal{L}_{\mathbb{X}}(\mathbb{X})$ is convex in $\mathbb{X}$.

Define the qn dimensional vector

$$\bar{X} = \{X_1(\beta_1), \ldots, X_1(\beta_q), \ldots, X_n(\beta_1), \ldots, X_n(\beta_q)\}.$$

So the gradient of $\mathcal{L}_{\mathbb{X}}(\mathbb{X})$ is defined by the $(qn \times 1)$ dimensional vector,

$$G_X = \left\{ G_{X_1(\beta_1)}, \ldots, G_{X_1(\beta_q)}, \ldots, G_{X_n(\beta_1)}, \ldots, G_{X_n(\beta_q)} \right\},$$

where

$$G_{X_i(\beta)} = \frac{\partial \mathcal{L}_{\mathbb{X}}(\mathbb{X})}{\partial X_i(\beta)} = s^* \left(\frac{D'}{L^2} X_i(\beta) - \frac{\pi_i(\beta) e^{-s^* X_i(\beta)}}{\sum_{\beta \in F} \pi_i(\beta) e^{-s^* X_i(\beta)}} \right). \tag{12.3}$$

The second derivatives are given by

$$\frac{1}{s^*} \frac{\partial^2 \mathcal{L}_{\mathbb{X}}(\mathbb{X})}{\partial X_i^2(\beta)} = \frac{D'}{L^2} + s^* \pi_i(\beta) e^{-s^* X_i(\beta)} \frac{\left(\sum_{\beta \in F} \pi_i(\beta) e^{-s^* X_i(\beta)} \right) - \pi_i(\beta) e^{-s^* X_i(\beta)}}{\left(\sum_{\beta \in F} \pi_i(\beta) e^{-s^* X_i(\beta)} \right)^2},$$

$$\frac{1}{s^*} \left. \frac{\partial^2 \mathcal{L}_{\mathbb{X}}(\mathbb{X})}{\partial X_i(\beta_1) \partial X_i(\beta_2)} \right|_{\beta_1 \neq \beta_2} = -s^* \frac{\pi_i(\beta_1) \pi_i(\beta_2) e^{-s^*(X_i(\beta_1) + X_i(\beta_2)}}{\left(\sum_{\beta \in F} \pi_i(\beta) e^{-s^* X_i(\beta)} \right)^2}, \quad \text{and}$$

$$\left. \frac{1}{s^*} \frac{\partial^2 \mathcal{L}_{\mathbb{X}}(\mathbb{X})}{\partial X_i(\beta_1) \partial X_j(\beta_2)} \right|_{\beta_1 \neq \beta_2, i \neq j} = 0.$$

Define the $q \times q$ matrix, H_{X_i}, such that for $a, b = 1, 2, \ldots, q$,

$$[H_{X_i}]_{a,b} = \frac{\partial^2 \mathcal{L}_{\mathbb{X}}(\mathbb{X})}{\partial X_i(\beta_a) \partial X_i(\beta_b)},$$

then using the above second order derivatives

$$H_{X_i} = s^* \left(\frac{D'}{L^2} I_q + \frac{s^*}{(J^T z_i)^2} \left((J^T z_i) \mathrm{Diag}(z_i) - z_i z_i^T \right) \right),$$

where $z_i = \left\{ \pi_i(\beta_a) e^{-s^* X_i(\beta_a)}, \ \ a = 1, \ldots, q \right\}$ is a $(q \times 1)$ vector, J is the all ones q vector and $\mathrm{Diag}(z)$ is the diagonal matrix with the elements of z on the diagonal. The Hessian of $\mathcal{L}_{\mathbb{X}}(\mathbb{X})$ defined by

$$[H_X]_{a,b} = \frac{\partial^2 \mathcal{L}_{\mathbb{X}}(\mathbb{X})}{\partial \bar{X}(a) \partial \bar{X}(b)}$$

is thus given by the block diagonal matrix

$$H_X = \mathrm{Diag}(H_{X_1}, H_{X_2}, ..., H_{X_n}). \tag{12.4}$$

Let v_i be any q vector,

$$\Psi_i = \left\{ \sqrt{z_i(1)}, \ldots, \sqrt{z_i(q)} \right\}^T \quad \text{and} \quad \Phi_i = \left\{ v_i(1)\sqrt{z_i(1)}, \ldots, v_i(q)\sqrt{z_i(q)} \right\}^T,$$

then

$$v_i^T H_{X_i} v_i = s^* \left(\frac{D'}{L^2} v_i^T v_i + \frac{s^*}{(J^T z_i)^2} \left((\Psi_i^T \Psi_i)(\Phi_i^T \Phi_i) - (\Psi_i^T \Phi_i)^2 \right) \right), \tag{12.5}$$

By the Cauchy-Schwartz inequality,

$$(\Psi_i^T \Psi_i)(\Phi_i^T \Phi_i) - (\Psi_i^T \Phi_i)^2 \geq 0,$$

and by substituting in (12.5) it follows that

$$v_i^T H_{X_i} v_i \geq \frac{s^* D'}{L^2} v_i^T v_i \geq 0, \tag{12.6}$$

where the last inequality is due to the fact that $s^* \geq 0$ and $v_i^T v_i = \|v_i\|^2 \geq 0$ for any vector v_i. If $s^* > 0$, then $v_i^T H_{X_i} v_i > 0$ for any nonzero vector v_i which implies that H_{X_i} is *positive definite*. Let $v = \left\{ v_1^T, v_2^T, ..., v_n^T \right\}^T$ be an arbitrary qn vector, then from (12.6) and (12.4), it follows that

$$v^T H_X v = \sum_{i=1}^{n} v_i^T H_{X_i} v_i \geq 0,$$

which proves that $\mathcal{L}_{\mathbb{X}}(\mathbb{X})$ is convex. Generally, $s^* > 0$ which would imply that H_X is *positive definite* and thus $\mathcal{L}_{\mathbb{X}}(\mathbb{X})$ is *strongly convex*. In this analysis, we assumed that $s = s^*$ since we will optimize for s and then for $\mathbb{X}$. However, for another $s \geq 0$, the term D' in (12.3) could be treated as another positive quantity and the analysis holds.

13. Iterative Algorithm

The proposed iterative algorithm for finding $\mathbb{X}^* = (X_i(\beta))$, and thus the optimum multiplicity matrix, could be summarized as follows:

ALGORITHM 13.1. *Let s^j and $\mathbb{X}^j = (X_i^j(\beta))$ be the values of s and $\mathbb{X}$ at the j^{th} iteration respectively. $\epsilon \approx 10^{-5}$ is a small number greater than zero.*

Initialize $\mathbb{X}^o = \frac{L^2}{D'}\Pi$, $s^o = 0.1 * \frac{D'}{L^2}$ *and* $j = 0$.

Do

$j := j + 1$

`I`. ***Solve*** *for s^j, (Eq. 11.3),*

$$\nabla_s(\mathcal{L}^*(s, \mathbb{X}^{j-1}) = \left.\frac{\partial \mathcal{L}^*(s, \mathbb{X}^{j-1})}{\partial s}\right|_{s=s^j} = 0$$

If s^j is negative then set s^j to be zero.

`II`. ***Solve*** *for $\mathbb{X}^j$, (Eq. 11.7),*

$$\nabla_{\mathbb{X}}(\mathcal{L}^*(s^j, \mathbb{X})) = \left.\left\{\frac{\partial \mathcal{L}^*(s^j, \mathbb{X})}{\partial X_i^j(\beta)}\ , i = 1, ..., n,\ \beta \in F\right\}\right|_{\mathbb{X}=\mathbb{X}^j} = 0$$

While

$$\left\|\frac{s^j - s^{j-1}}{s^{j-1}}\right\|_1 \leq \epsilon.$$

For the case of finite costs, the optimized integer multiplicity matrix, $M = (m_i(\beta))$ is found from the optimized matrix $\mathbb{X}^* = (X_i^*(\beta))$ by the inverse transformation,

$$m_i(\beta) = \text{Round}\left\{\max\left\{0, X_i^*(\beta) - 0.5\right\}\right\}, \tag{13.1}$$

where Round $\{\}$ *is the rounding to the nearest integer.*

In our implementation and for the simulation results in this paper, we replace the command **Solve** by a Newton type algorithm. Other algorithms such as the gradient descent algorithm, which is less computationally complex, were also tested. However, the Newton algorithm described in appendix A, achieved the best results. Since Step `II` in the above algorithm is an optimization in qn variables, the entries of $\mathbb{X}$, it is computationally expensive. However, the computational complexity could be reduced dramatically by observing that the entries of each column in Π, π_i, sum to one, and that for reasonable operating signal to noise ratios (SNRs) only a small fraction of them have a relevant value while the rest tend to be negligible or zero. Thus, in optimizing for $\mathbb{X}$ only the elements $X_i(\beta)$ corresponding to elements $\pi_i(\beta)$ above a certain threshold are considered for optimization while the others are set to zero. Practically, this threshold could be set to 10^{-6} or 10^{-7}. This implies that the complexity of our algorithm decreases with an increase in the operating SNR, which is usually the case for operating conditions.

14. Numerical Results

In this section we will refer to our method as the Chernoff method. The Gaussian approximation of [**18**] is referred to as the Gauss method and the Kötter-Vardy algorithm, (6.1), as KV. A hard decision bounded minimum distance decoder, as the Berlekamp-Massey algorithm, is referred to as BM. It is to be noted that we used the condition of (3.1) to test if the transmitted codeword is on the GS generated list for all ASD algorithms compared. If the sufficient condition is satisfied then a

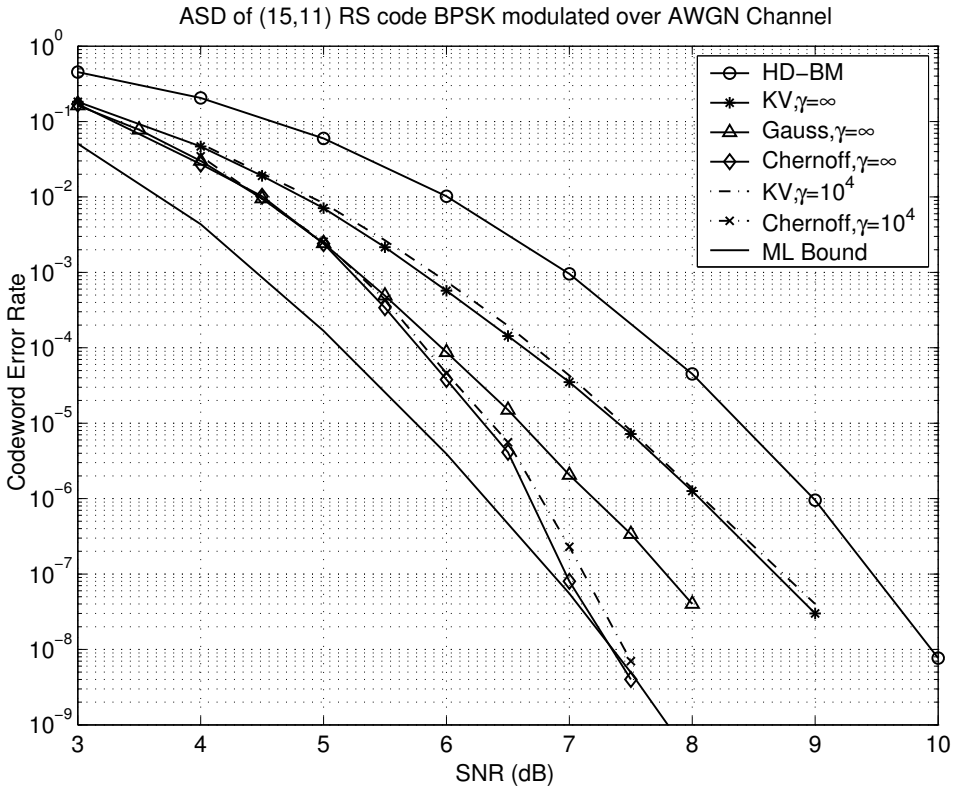

FIGURE 2. A comparison of the performance of ASD algorithms when decoding an $(15, 11)$ RS code BPSK modulated over an AWGN channel. Their performance is also compared to an averaged upper bound on the performance of the ML decoder.

decoding success is signaled. This is somehow justified by the fact that, on average, the list size is one [**14**]. If the GS generated list is empty, then the condition will not be satisfied, and a decoder error is signaled. In a real time implementation, if more than one codeword is on the generated list, then the most reliable codeword (with respect to the soft output from the channel) is chosen as the decoder output.

To test our theories, we simulated the performance of the $(15, 11)$ RS code over the finite field F of 16 elements, $GF(16)$, on an additive white gaussian noise (AWGN) channel. These results are shown in Fig. 2 and Fig. 3 for the cases of binary phase shift keying (BPSK) and 16-ary phase shift keying (PSK) modulation schemes respectively.

We see that the Chernoff technique shows a marked superiority when compared to the KV technique, for both finite and infinite cost matrices. For BPSK modulation, infinite cost γ, and an error rate of 4×10^{-8}, our algorithm has about 0.9 dB, 1.8 dB and 2.5 dB coding gains over the Gauss, KV and BM algorithms respectively. Simulation results for a finite cost of 10^4 also show the potential of our algorithm over previously proposed ones. A tight averaged upper bound on the maximum likelihood error probability [**3**] is also plotted. Since it is the binary

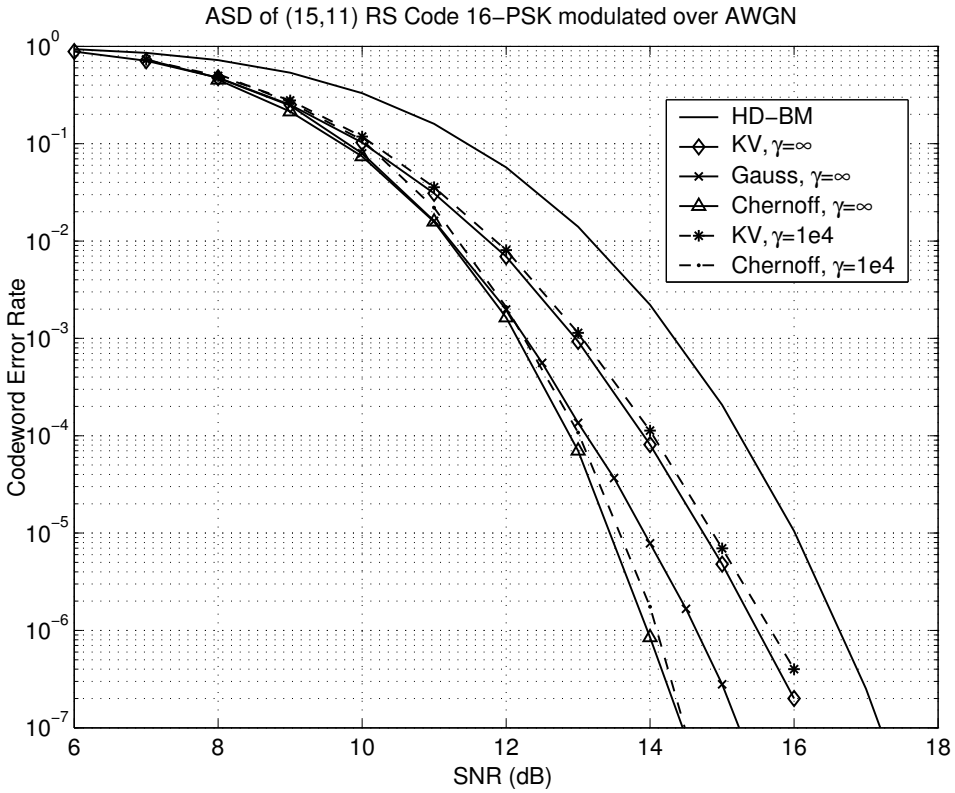

FIGURE 3. Performance curves for decoding an $(15, 11)$ RS code, 16-PSK modulated over an AWGN channel, using different ASD algorithms.

image of the RS code which is modulated and transmitted over the channel, and the binary image is not unique but depends on the basis used to represent the symbols in $GF(16)$ as bits, this bound was derived by averaging over all possible binary images over an RS code. By comparing with actual simulations for maximum likelihood decoding of the $(15, 11)$ RS code over an AWGN channel this bound was shown to be tight [**3**]. Our algorithm has a near-ML performance at high signal to noise ratios.

Similarly, for the case of 16-ary PSK, the Chernoff algorithm has about 2.6 dB gain over the BM algorithm at a codeword error rate of 10^{-7}. The performance gain over KV is about 1.7 dB at an error rate of 10^{-6}.

Numerical results for ASD decoding of the $(31, 25)$ RS code over $GF(32)$ BPSK modulated over AWGN channel are shown in Figure 4. As seen the Chernoff algorithm has up to 2 dB gain over the hard-decision BM algorithm. The coding gain over the KV algorithm and the Gaussian approximation increases at the tail of error probability. The averaged bound on the ML error probability is also plotted. It is observed that that at high SNRs, our algorithm is near optimal.

To demonstrate the convergence of our proposed algorithm, we plot the value of s^j, (see Algorithm 13.1), versus the iteration number j for a fixed value of

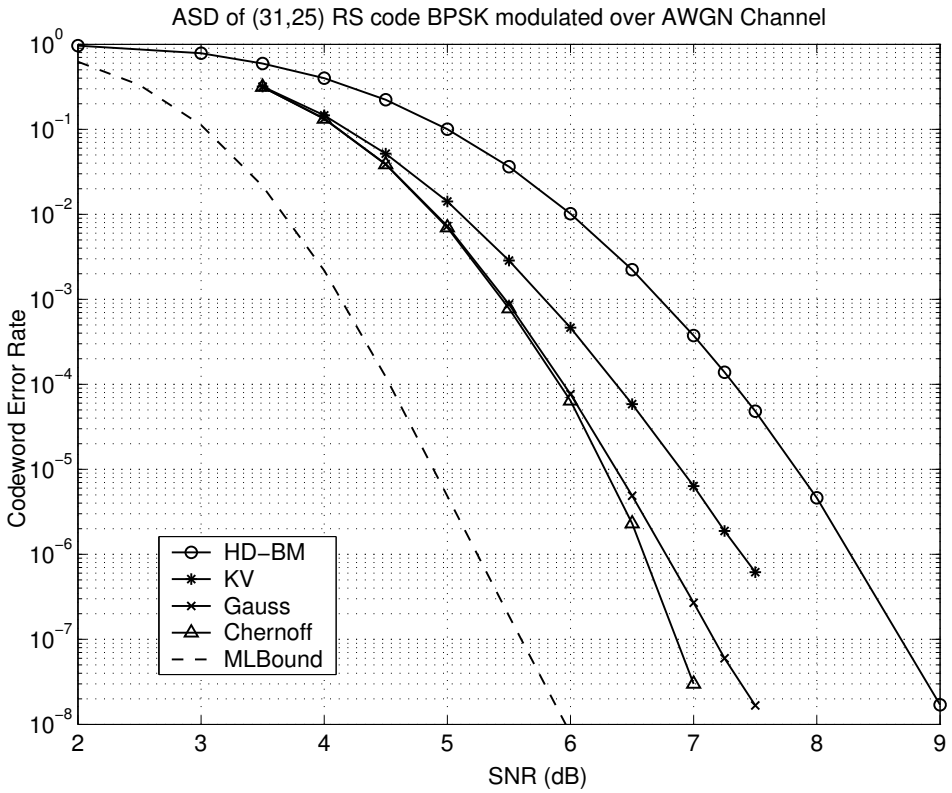

FIGURE 4. An (31,25) RS code is BPSK modulated over an AWGN channel. ASD algorithms are compared at infinite interpolation costs. The Chernoff algorithm has a better performance than the Gauss and KV algorithms. The performance curve of a bounded minimum distance decoder and an averaged upper bound on the performance of the ML decoder are also plotted.

SNR. This is shown in Figure 5 for a randomly transmitted (15,11) RS codeword and BPSK modulation with an SNR of 6 dB. The average codeword error rate is plotted in Figure 6 versus the number of iterations at a SNR of 5.5 dB. These figures demonstrate the fast convergence of the algorithm in terms of the number of (global) iterations.

The performance gains of our algorithm over that of the Gaussian approximation could be reasoned by observing that the Gaussian approximation finds the multiplicity matrix of infinite cost that minimizes the error probability assuming that the score has a Gaussian distribution.It could be shown that this is equivalent to minimizing the Chebychev bound [**20, 25**] on the error probability assuming that the score is symmetrically distributed around its mean. It is well known that the Chernoff bound is a tighter upper bound than the Chebychev bound (cf. [**20, 25**]). Further more, no assumptions about the distribution of the score is made in deriving our algorithm.

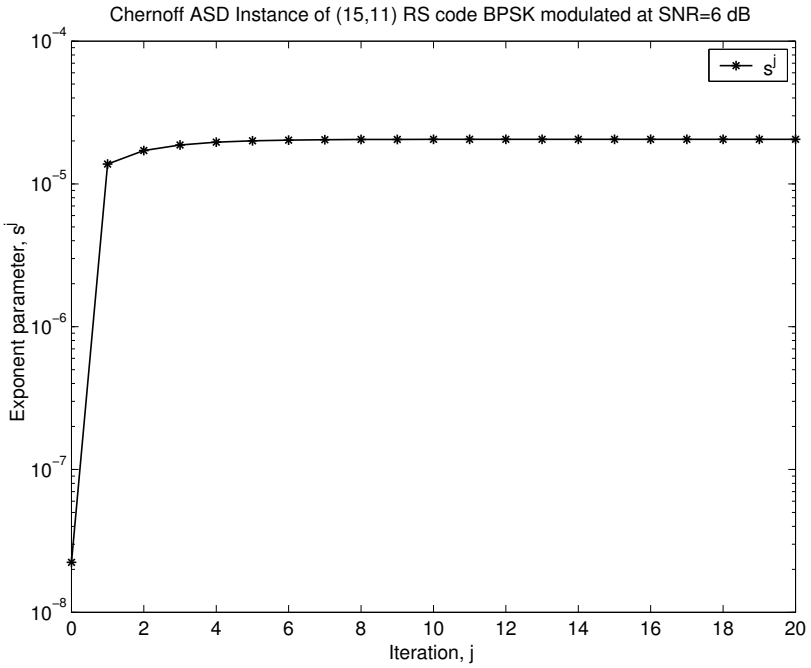

FIGURE 5. A decoding instance of the $(15, 11)$ RS code, BPSK modulated over an AWGN channel at a fixed SNR of 6 dB, using Chernoff ASD. The convergence of the algorithm is conveyed by the fast adaptation of the exponential parameter s^j.

It is observed that the coding gains of the Chernoff algorithm, developed in this paper, over other ASD algorithms increases as the SNR increases and approaches that of the ML bound. This somehow proves the conjecture that our algorithm is optimal at the tail of error probability. The reasoning behind that is the fact that the Chernoff bound, in general, is an exponentially tight upper bound at the tail of error probability and closely approximates the true error probability. In another way, this shows the potential of using the Chernoff algorithm in favorable operating conditions.

15. Conclusions

The goal of this paper was to find the ultimate capabilities of algebraic soft decoding of Reed-Solomon codes. Since the performance of ASD depends mainly on the interpolation multiplicities assigned, we explored a novel multiplicity assignment algorithm that results in an improved performance. The multiplicity assignment algorithm proposed aims at directly minimizing the decoding error probability. Reasonable approximations and relaxations were made to simplify the problem. However, since the actual error probability is relatively hard to compute, we aimed at finding the multiplicity matrix that will minimize an upper bound (the Chernoff bound) on the error probability. We explore the cases of both finite and infinite cost multiplicity matrices. The problem is formulated as a constrained optimization problem and an iterative algorithm is developed that will find the optimum

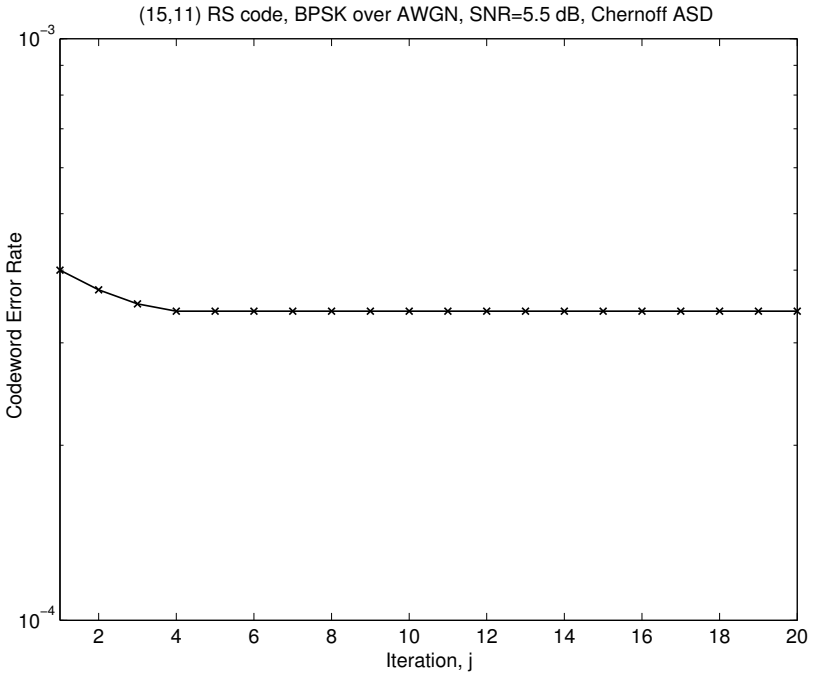

FIGURE 6. The convergence of the Chernoff ASD algorithm is demonstrated by plotting the average codeword error probability versus the number of iterations at a fixed SNR of 5.5 dB.

multiplicity matrix. Numerical results show that our algorithm is superior to other multiplicity assignment algorithms found in the literature.

Topics of future research include finding algorithms that will directly minimize the decoding error probability. A future direction is to use belief propagation type algorithms [**6, 15**] to precondition the channel reliability matrix before passing it to the multiplicity assignment algorithm. The initial results of this research direction are promising [**4**].

Appendix A: The Newton Algorithm

We briefly sketch the Newton algorithm used to minimize an arbitrary function $f(\mathbf{x})$ in m variables. For more details, we refer the reader to [**2**] and [**10**]. The gradient of $f(\mathbf{x})$ is the $(m \times 1)$ dimensional vector $\nabla f(\mathbf{x})$, and its $(m \times m)$ Hessian is $H_f(\mathbf{x})$. We assume that $f(\mathbf{x})$ is twice continuously differentiable, there exists at least one solution $\mathbf{x_{opt}}$ such that $\nabla f(\mathbf{x_{opt}}) = 0$ and the Hessian $H_f(\mathbf{x})$ is positive definite for $\mathbf{x} = \mathbf{x_{opt}}$.

Let $\mathbf{x_o}$ be the initial iterate, then for iteration n:

1. Test for termination:

Stop if $\|\nabla f(\mathbf{x_n})\| \leq \tau_r \|\nabla f(\mathbf{x_o})\| + \tau_a$, τ_r and τ_a are small positive numbers and are called the relative tolerance and absolute tolerance respectively.

2. Find the Newton Direction, d:

Calculate the Hessian, $H_f(\mathbf{x_n})$ if an analytical expression is found, otherwise approximate $H_f(\mathbf{x_n})$ with a finite difference Hessian. The later case involves m

new evaluations, $\nabla f(\mathbf{x_n} + \delta \mathbf{e_j}),\ j = 1, ..., m$ where $\mathbf{e_j}$ is the unit vector in the j^{th} coordinate direction. The Newton direction satisfies

$$H_f(\mathbf{x_n})\mathbf{d} = -\nabla f(\mathbf{x_n})$$

This requires the LU factorization of the Hessian using Gaussian elemination, $H_f(\mathbf{x_n}) = PLU = L'U$, and solving for $L'z = -\nabla f(\mathbf{x})$ and $U\mathbf{d} = z$. The LU decomposition require $m^3 + O(m^2)$ flops and solving for the triangular systems requires $m^2 + O(m)$ flops. The complexity of the algorithm lies here.

3. Line Search:

The Armijo rule for calculating the length of the Newton step, λ, iteratively finds $\lambda_o, \lambda_1, ...\lambda_k$ till

$$\|\nabla f(\mathbf{x_n} + \lambda_k \mathbf{d})\| < (1 - \alpha\lambda_k)\|\nabla f(\mathbf{x_n})\|$$

for the smallest $k \geq 0$ and $\alpha \in (0, 1)$ is typically 10^{-4} to easily satisfy the equation. One method is to let $\lambda_o = 1$ and $\lambda_k = \lambda_{k-1}/2$ for $k \geq 1$. In this implementation, λ_{k+1} is the minimizer of the parabola fitted to the points $\phi(0), \phi(\lambda_k)$ and $\phi(\lambda_{k-1})$ on the interval $[\lambda_k/10, \lambda_k/2]$ where $\phi(\lambda) = \|\nabla f(\mathbf{x} + \lambda\mathbf{d})\|^2$.

4. Update x:

$$\mathbf{x_{n+1}} = \mathbf{x_n} + \lambda\mathbf{d}.$$

Since the Hessian is computationally excessive to compute and factor, a hybrid Chord-Newton strategy is used; the Hessian is updated only after a certain number of nonlinear iterations or if the ratio of successive norms of the nonlinear residuals $\|\nabla f(\mathbf{x_n})\|/\|\nabla f(\mathbf{x_{n-1}})\|$ is larger than a certain threshold, i.e. the rate of decrease in the residual is not sufficiently rapid.

References

[1] L. Bahl, J. Cocke, F. Jeinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate." *IEEE Trans. Inform. Theory*, vol. 20, pp. 284–7, Mar 1974.

[2] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge: Cambridge U. Press, 2004.

[3] M. El-Khamy and R. J. McEliece, "Bounds on the average binary minimum distance and the maximum likelihood performance of Reed Solomon codes," in *42nd Allerton Conf. on Communication, Control and Computing*, 2004.

[4] M. El-Khamy and R. J. McEliece, "Iterative algebraic soft decision decoding of Reed-Solomon codes," in *Proc. of International Symposium of Information Theory and its Applications, Parma, Italy, 2004.*

[5] M. El-Khamy, R. J. McEliece, and J. Harel, "Performance enhancements for algebraic soft decision decoding of Reed-Solomon codes," in *Proc. of International Symosium of Information Theory, Chicago, IL, USA, 2004.*

[6] R. Gallager, *Low Density Parity Check Codes.* MIT Press, 1963.

[7] W. J. Gross, F. R. Kschischang, R. Kötter, and P. G. Gulak, , "Applications of algebraic soft-decision decoding of Reed-Solomon codes," submitted to the *IEEE Trans. Commun.*, preprint dated July 23, 2003.

[8] V. Guruswami and M. Sudan, "Improved decoding of Reed-Solomon codes and algebraic geometry codes," *IEEE Trans. Inform. Theory* vol. 45 no. 6 (Sept. 1999), pp. 1757–1767.

[9] J. Hagenauer and P. Hoher, "A Viterbi algorithm with soft-decision outputs and its applications," in *GLOBECOM'89, Dallas, Texas*, 1989, pp. 47.1.1–47.1.7.

[10] C. T. Kelley, *Solving Nonlinear Equations with Newton's Method*, Society for Industrial and Applied Mathematics, Philadelphia, 2003.

[11] R. Kötter and A. Vardy, "Algebraic soft-decision decoding of Reed-Solomon codes," *IEEE Trans. Inform. Theory* vol. IT-49 no. 11 (Nov. 2003), pp. 2809–2825.

[12] R. J. McEliece, *The Guruswami-Sudan Algorithm for Decoding Reed-Solomon Codes.* JPL IPN Progress Report 42-153 (May 15, 2003). Available at `http:// ipnpr. jpl. nasa. gov/ progress_report/42-153/`.

[13] R. J. McEliece, *The Theory of Information and Coding*, 2nd ed. Cambridge: Cambridge U. Press, 2002.
[14] R. J. McEliece, "On the average list size for the Guruswami-Sudan decoder," in *ISCTA03*, 2003
[15] R. McEliece, D. MacKay, and J. Cheng, "Turbo decoding as an instance of pearl's belief propagation algorithm," *IEEE J. Select. Areas Commun.*, vol. 16, pp. 140–152, Feb 1998.
[16] R. Nielsen and T. Hoeholdt, "Decoding Reed-Solomon codes beyond half the minimum distance," pp. 221–236 in *Cryptography and Related Areas*, J. Buchmann, T. Hoeholdt, H. Stichenoth, and H. Tapia-Recillas, eds. Springer-Verlag, 2000.
[17] R. R. Nielson, "Decoding concatenated codes with Sudan's algorithm", submitted to *IEEE Trans. Inform. Theory.*
[18] F. Parvaresh and A. Vardy, "Multiplicity asssignments for algebraic soft-decision decoding of Reed-Solomon codes,", in *Proc. ISIT 2003*, pp. 205.
[19] L. Pecquet, "List decoding of algebraic-geometric codes," PhD thesis, University Paris, December 2001.
[20] J. G. Proakis, *Digital Communications*, 4th ed. McGraw-Hill, 2001.
[21] N. Ratnakar and R. Kötter, "Exponential error bounds for algebraic soft-decision decoding of Reed-Solomon codes," submitted to *IEEE Trans. Inform. Theory.*
[22] I. S. Reed and G. Solomon, "Polynomial codes over certain finite fields," *J. Soc. Industrial Appl. Math.*, vol. 8 (1960), pp. 300-304.
[23] M. Sudan, "Decoding of Reed Solomon codes beyond the error-corrrection bound," *J. Complexity*, vol. 13 (1997), pp. 180–193.
[24] A. Viterbi, "Error bounds on convolutional codes and an assymptotically optimum decoding algorithm," *IEEE Trans. Inform. Theory*, vol. 13, pp. 260–269, April 1967.
[25] J. M. Wozencraft and I. M. Jacobs,"*Principles of Communication Engineering*," John Wiley & Sons, Inc., 1965.

California Institute of Technology, 1200 E. California Blvd., Pasadena, CA 91125, USA
E-mail address: mostafa@systems.caltech.edu

California Institute of Technology, 1200 E. California Blvd., Pasadena, CA 91125, USA
E-mail address: rjm@systems.caltech.edu