FISHER INFORMATION DECISION DIRECTED DISCRETE OPTIMISATION

Ian Brace and Jonathan H. Manton

Dept of Electrical and Electronic Engineering
University of Melbourne, Victoria, 3010, Australia
i.brace@ee.mu.oz.au

ABSTRACT

Finite alphabet optimisation problems occur in many fields of engineering, including wireless communications and blind source separation. An optimal solution through exhaustive search is often computationally intractable, so sub-optimal solutions are employed. One popular approach is to simply round each element of the unconstrained solution to the nearest member of the known alphabet. This paper presents a novel approach which has better error performance than rounding but with only a moderate increase in complexity. The method uses Fisher Information to determine the order in which optimisation is carried out. The inverse of the Fisher Information Matrix indicates which element of the estimate is, on average, most likely to have the smallest error. Thus the first element to be optimised is the one most likely to be correct. This then improves the likelihood of subsequent elements being correct. The method is developed and an example is included of its application to the discrete blind source separation problem.

1. INTRODUCTION

Finite alphabet optimisation problems occur in many fields of engineering, including wireless communications [3], multi-user detection [6] and blind source separation [2], [5]. Generally, the aim is to determine a solution to a parameter vector, where the elements of the vector are constrained to lie in a finite alphabet. In theory, the finite alphabet problem should be simpler to solve than the unconstrained case, since one may conduct an exhaustive search of all possible permutations of the parameter vector from the finite alphabet and select the permutation which produces the smallest cost value. Whilst this approach would have excellent error performance, in practice it is computationally intractable in all but the simplest of cases. As the dimension of the parameter vector increases the number of permutations increases exponentially, making the exhaustive search impractical even for simple binary alphabets. A popular alternative is to perform a computationally simpler unconstrained optimisation and then round the results to the nearest element of the finite alphabet. Whilst this is practical, it has relatively poor error performance.

Decision directed optimisation refers to a class of algorithms in which early decisions are used to influence later decisions. In such cases the *order* in which the parameters are decoded is a powerful degree of freedom which can be exploited. For example, in multi-user wireless communications the strongest user signal may be decoded first, with the remaining signals regarded as noise. The recovered signal is then re-encoded and subtracted from the original received signal. The next strongest signal is then decoded, treating the remaining signals as noise, and the procedure is repeated [7]. Because early errors will propagate through the decoding process the order in which elements are decoded is critical. This idea is extended in [6], where detection is no longer assumed to be in the order of decreasing signal energy. The algorithms in [6] rely on the effective energy in the signal as the signal to noise ratio becomes very large and so the complete $K!$ possible decoding orders (for K users) is considered and the order of detection is determined by an asymptotic energy metric.

In [3], Manton proposed a different decision directed strategy using Fisher Information as the ordering metric. The general method will be developed in section two. It is called Fisher Information Decision Directed optimisation because it relies on the inverse of the Fisher Information Matrix (ie, the Cramer-Rao Lower Bound (CRLB)) to determine the best candidate element to be optimised next. Fisher Information provides a mechanism for deciding which element of a parameter vector is, on average, most likely to be correct. This allows us to iteratively fix the vector elements, *beginning* with the one most likely to be correct. This paper then extends these ideas by investigating the results of fixing more than one bit at a time. It has been shown in [3] that by improving the likelihood of initial correct decisions, the likelihood of subsequent correct decisions is also improved. In section three a weighted probability of bit error is used to measure the balance between improved error performance and increased complexity and it is shown that for a binary alphabet a minimum of two elements should be fixed simultaneously. Section four provides an application example which applies this technique to a finite alphabet blind source separation problem. Fisher Information Decision Directed optimisation provides an optimisation scheme which can be applied across the complete range of finite alphabet problems.

2. FISHER INFORMATION DECISION DIRECTION

2.1. Fisher Information

Consider an observation vector $\mathbf{y} = \Phi(\mathbf{s}, \mathbf{h}) + \mathbf{n}$ where $\mathbf{s} \in \Omega^n$ is an n-dimensional parameter vector with elements drawn from the finite alphabet Ω, $\mathbf{h}$ is a vector of real valued parameters which may or may not be known, $\mathbf{n}$ is a noise vector and $\mathbf{y}$ has a known probability density function $p(\mathbf{y}; \mathbf{s}, \mathbf{h})$. It is assumed that a suitable cost function exists, $g(\hat{\mathbf{s}}; \mathbf{y}, \mathbf{h})$, where $\hat{\mathbf{s}}$ is an estimate of $\mathbf{s}$, and has a minimum when $\hat{\mathbf{s}}$ equals $\mathbf{s}$. In order to determine which element of $\hat{\mathbf{s}}$ to quantise first we evaluate the

This work was supported by the University of Melbourne and the CRC for Sensor Signal and Information Processing.

Fisher Information Matrix [1]

$$I(\mathbf{s},\mathbf{h}) = \begin{bmatrix} -E\left[\frac{\partial^2 \ln p(\mathbf{y};\mathbf{s},\mathbf{h})}{\partial \mathbf{s}^2}\right] & -E\left[\frac{\partial^2 \ln p(\mathbf{y};\mathbf{s},\mathbf{h})}{\partial \mathbf{s}\partial \mathbf{h}}\right] \\ -E\left[\frac{\partial^2 \ln p(\mathbf{y};\mathbf{s},\mathbf{h})}{\partial \mathbf{h}\partial \mathbf{s}}\right] & -E\left[\frac{\partial^2 \ln p(\mathbf{y};\mathbf{s},\mathbf{h})}{\partial \mathbf{h}^2}\right] \end{bmatrix} \quad (1)$$

where $E[\,]$ is the expectation operator with respect to the noise, $\mathbf{n}$, and the derivative is with respect to the true value of $\mathbf{s}$ and $\mathbf{h}$. It can be shown that under a zero mean white Gaussian noise assumption the inverse of the Fisher Information Matrix is given by [4]

$$I(\mathbf{s},\mathbf{h})^{-1} = \sigma^2 \begin{bmatrix} \left[G_1^T Q_2 G_1\right]^{-1} & \star \\ \star & \left[G_2^T Q_1 G_2\right]^{-1} \end{bmatrix} \quad (2)$$

where $G_1 = \frac{\partial \Phi}{\partial \mathbf{s}}$, $G_2 = \frac{\partial \Phi}{\partial \mathbf{h}}$, and $Q_2 = I - G_2\left(G_2^T G_2\right)^{-1} G_2^T$. Note, however, that we need not calculate the Fisher Matrix with respect to $\mathbf{h}$ if $\mathbf{h}$ is real valued. Thus, the inverse of the Fisher Information Matrix with respect to $\mathbf{s}$ is the upper left block of (2). If $\mathbf{s}$ is the only unknown parameter then (2) reduces to

$$I(\mathbf{s})^{-1} = \sigma^2 \left[G_1^T G_1\right]^{-1} \quad (3)$$

If the estimator $\hat{\mathbf{s}}$ is Fisher efficient (unbiased and attains the CRLB) then the smallest value on the leading diagonal of the inverse of the Fisher Information Matrix will indicate which element of $\hat{\mathbf{s}}$ is likely to have the smallest error component. ie, $I(\mathbf{s})_{i,i}^{-1} = E\left[(\hat{\mathbf{s}}_i - \mathbf{s}_i)^2\right]$ Thus, if the *ith* element of the leading diagonal is the smallest element on the diagonal then the *ith* element of $\hat{\mathbf{s}}$ should be fixed first.

2.2. Fixing Process

Because $\mathbf{s}$ is drawn from a finite alphabet the *ith* element of $\hat{\mathbf{s}}$ is set to the first alphabet value, Ω^1. The remaining elements of $\hat{\mathbf{s}}$ can take on continuous real values and the cost function $g(\hat{\mathbf{s}};\mathbf{y},\mathbf{h})$ is mimimised. The *ith* element of $\hat{\mathbf{s}}$ is set to the next alphabet value, Ω^2, and the process is repeated. Once all alphabet values have been tried, the *ith* element of $\hat{\mathbf{s}}$ is fixed to the alphabet value which produced the lowest cost function result. This element is now locked in and does not change for the remainder of the optimisation process. We now compute the Fisher Information Matrix for the remaining unlocked elements of $\hat{\mathbf{s}}$, identify the next element to be tested and compute the cost function $g(\hat{\mathbf{s}};\mathbf{y},\mathbf{h})$ for each alphabet member. This second element is then locked and we proceed to the third element, etc. This leads to Algorithm 1.

If the alphabet has k values then the optimisation procedure must be carried out kn-times. We may generalise this procedure by choosing to optimise more than one element of $\hat{\mathbf{s}}$ at each iteration. For example, if we choose to optimise two elements at a time then the procedure is more accurate, but must be carried out $\frac{k^2 n}{2}$ times. If we optimise p elements at a time then the procedure must be carried out $\frac{k^p n}{p}$ times. If we optimise n elements at a time we return to enumerating all possible solutions, which returns an optimal result but requires the procedure to be carried out k^n times.

In general the Fisher Information Matrix may depend on $\mathbf{y}$, $\mathbf{h}$ and $\mathbf{s}$. In such circumstances the true value of $\mathbf{h}$ and $\mathbf{s}$ may not be known and an initial estimate of each parameter vector must be used to calculate the Fisher Information Matrix. The accuracy

Algorithm 1 Fisher Information Decision Directed Optimisation

1) Consider an observation vector $\mathbf{y} = \Phi(\mathbf{s},\mathbf{h}) + \mathbf{n}$. The vector $\mathbf{y} \in \mathbb{R}^m$, $\mathbf{s} \in \Omega^n$ is an n-dimensional source vector with elements drawn from the finite alphabet Ω, $\mathbf{h}$ is a real valued parameter vector and $\mathbf{n} \in \mathbb{R}^m$ is a random Gaussian noise vector. The least squares solution is to find the vector $\hat{\mathbf{s}} \in \Omega^n$ which minimises $g(\hat{\mathbf{s}};\mathbf{y},\mathbf{h})$
2) Evaluate the inverse of the Fisher Information Matrix where the dimension of $\mathbf{s}$ is the number of unlocked elements in $\hat{\mathbf{s}}$. $I(\mathbf{s})^{-1} = \left[G_1^T Q_2 G_1\right]^{-1}$, where $G_1 = \frac{\partial \Phi}{\partial \mathbf{s}}$, $G_2 = \frac{\partial \Phi}{\partial \mathbf{h}}$, and $Q_2 = I - G_2\left(G_2^T G_2\right)^{-1} G_2^T$. Determine $i = \arg\min_i I(\mathbf{s})_{i,i}^{-1}$.
3) Set $\hat{\mathbf{s}}_i = \Omega^j$ where the number of elements in the alphabet are $1 \ldots j \ldots N$. Optimise the remaining real-valued elements of $\hat{\mathbf{s}}$ and evaluate $g_j(\hat{\mathbf{s}};\mathbf{y},\mathbf{h})$. The notation g_j represents the cost function result when the alphabet member Ω^j is used.
4) Repeat step 3 for all N elements of Ω.
5) $k = \arg\min_j g_j(\hat{\mathbf{s}};\mathbf{y},\mathbf{h})$. Set $\hat{\mathbf{s}}_i = \Omega^k$and lock.
6) Repeat step 2 until all elements of $\hat{\mathbf{s}}$ are locked.

of the estimate will affect the accuracy of the resulting matrix and hence the order in which the bits are optimised. However, the resulting order is unlikely to produce an error performance any worse than simply rounding the vector to the nearest finite alphabet member. If the estimate of $\mathbf{s}$ is close to the true $\mathbf{s}$ then the error performance will be considerably better than simple rounding.

3. COMPLEXITY STUDY

In order to examine the trade off between error performance and complexity the procedure described in section two was simulated using the simple observation vector

$$\mathbf{y} = A\mathbf{s} + \mathbf{n} \quad (4)$$

where $\mathbf{y} \in \mathbb{R}^{12}$ is the observation vector, $\mathbf{s} \in \Omega^{12}$ is a 12-dimensional source vector with elements drawn from the finite alphabet $\Omega = \{\pm 1\}$, $A \in \mathbb{R}^{12\times 12}$ is a known, real valued matrix and $\mathbf{n} \in \mathbb{R}^{12}$ is a random Gaussian noise vector. The least squares solution is to find the vector $\hat{\mathbf{s}} \in \Omega^{12}$ which minimises

$$J(\hat{\mathbf{s}}) = \|\mathbf{y} - A\hat{\mathbf{s}}\|^2 \quad (5)$$

and the procedure followed is shown in Algorithm 2.

Figure 1 compares the error performance for a simple rounding algorithm and Fisher Information Decision Directed optimisation for several values of p. The Fisher method was tested optimising one, two, three and four simultaneous bits. Clearly, even a one bit Fisher scheme out performs the simple rounding approach, but if we are prepared to accept some increased complexity then the simultaneous optimisation of several bits at a time offers an even greater improvement in error performance.

Figure 2 shows the complexity, in terms of the number of iterations required, against the resultant probability of bit error for several signal to noise ratios. A complexity of one indicates the simple rounding method. Significantly, for a binary alphabet the complexity involved for $p = 1$ and $p = 2$ is the same, and is represented by the vertical line in figure 2. However, figure 1 clearly

Algorithm 2 Fisher Information Decision Directed Optimisation - $\mathbf{y} = A\mathbf{s} + \mathbf{n}$

1) Consider an observation vector $\mathbf{y} = A\mathbf{s} + \mathbf{n}$. The vector $\mathbf{y} \in \mathbb{R}^{12}$, $\mathbf{s} \in \Omega^{12}$ is a 12-dimensional source vector with elements drawn from the finite alphabet $\Omega = \{\pm 1\}$, and $\mathbf{n} \in \mathbb{R}^{12}$ is a random Gaussian noise vector. The least squares solution is to find the vector $\hat{\mathbf{s}} \in \Omega^{12}$ which minimises $J(\hat{\mathbf{s}}) = \|\mathbf{y} - A\hat{\mathbf{s}}\|^2$
2) Evaluate the inverse of the Fisher Information Matrix where the dimension of $\mathbf{s}$ is the number of unlocked elements in $\hat{\mathbf{s}}$. $I(\mathbf{s})^{-1} = \left[A^T A\right]^{-1}$. Determine $i = \arg\min_i I(\mathbf{s})^{-1}_{i,i}$.
3) Set $\hat{\mathbf{s}}_i = +1$. Optimise the remaining real-valued elements of $\hat{\mathbf{s}}$ and evaluate $J_1(\hat{\mathbf{s}}) = \|\mathbf{y} - A\hat{\mathbf{s}}\|^2$. The notation J_1 represents the cost function result when the alphabet member $\Omega = +1$ is used.
4) Repeat step 3 for $\Omega = -1$.
5) $k = \arg\min_j J_j(\hat{\mathbf{s}})$. Set $\hat{\mathbf{s}}_i = \Omega^k$and lock.
6) Repeat step 2 until all elements of $\hat{\mathbf{s}}$ are locked.

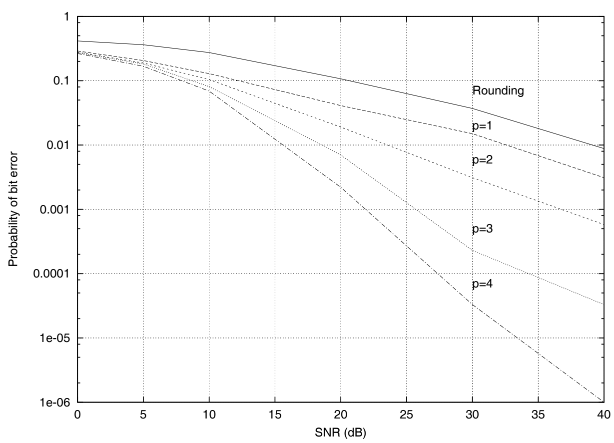

Fig. 1. Probability of bit error using the Fisher algorithm optimising 'p' bits simultaneously

shows that $p = 2$ offers an error performance improvement of up to 10 dB over the $p = 1$ scheme. Thus, for a binary alphabet, the Fisher Information Decision Directed algorithm should always be implemented by optimising a minimum of two bits per iteration as we obtain an error performance gain over the $p = 1$ scheme with no additional computational cost.

A natural question which arises from this method is at what point does the cost of the additional iterations outweigh the improvement in the bit error rate? One way to determine this is by using figure 2. If the anticipated signal to noise ratio environment is known then the smallest complexity can be chosen which will satisfy the required bit error rate. Alternatively, we can consider the question as a trade off between probability of bit error and complexity. In figure 3 the probability of bit error has been "normalised" for a signal to noise ratio of 10, 20 and 30 dB to create a weighted probability of bit error, where increased complexity degrades the weighted error. The weighted probability

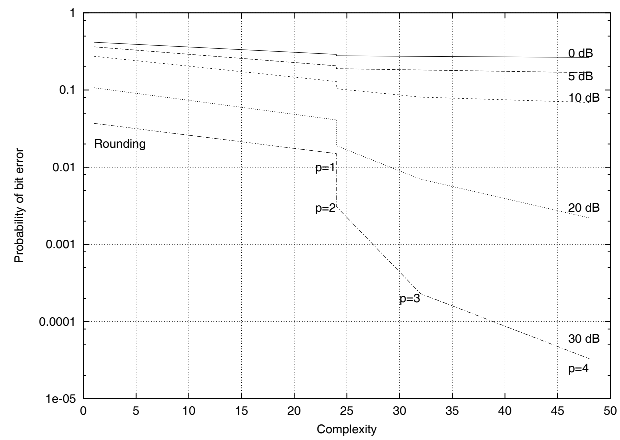

Fig. 2. Probability of bit error against the complexity of the Fisher algorithm in terms of the number of iterations. Curves are shown for 0, 5, 10, 20 and 30 dB. Note that for p=1 and p=2 the number of iterations required is the same, but p=2 offers a superior error performance.

of bit error, P_W, is given by

$$P_W(c) = P_e^{\frac{1}{1+\ln c}}$$

where $c \propto \frac{2^p n}{p}$ represents the complexity. The exact units used to measure complexity may vary with the application. In this case the measure of complexity was the number of iterations normalised to the $p = 1$ scheme. ie, $c = \frac{1}{24}\frac{2^p n}{p}$. Figure 3 shows that for low signal to noise ratios the improvement through Fisher Information Decision Directed optimisation is quickly offset by the increased complexity, such that the weighted probability of bit error increases after $p = 2$. However, at higher signal to noise ratios there continues to be a relative performance gain up to $p = 3$.

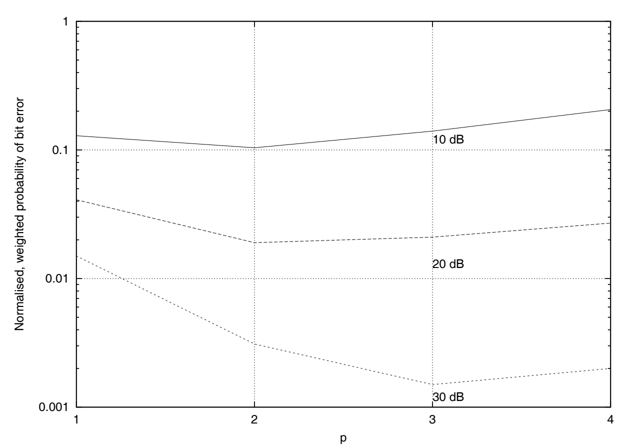

Fig. 3. Normalised probability of bit error weighted by the increase in iterations required.

4. BLIND SOURCE SEPARATION OF FINITE ALPHABET SIGNALS

Signal separation involves recovering several source signals when the only available information is a linear mix of the signals. This typically occurs in narrow band array processing, where the linear mixture results from the receiver array geometry. The exact linear mixture is often unknown because no assumptions are made about the array geometry. Thus, the received signal can be modelled as

$$Y = AS + N(0, \sigma^2) \tag{6}$$

where $S \in \mathbb{R}^{n \times N}$ is the source matrix and each row of S represents N samples of a source signal. The mixing matrix, $A \in \mathbb{R}^{m \times n}$, represents the geometry of the array and the observation matrix, $Y \in \mathbb{R}^{m \times N}$, is corrupted by Gaussian white noise. We are interested in finding the least squares estimate of A and S from the cost function

$$\phi(A) = \min_{S} \|Y - AS\|^2. \tag{7}$$

An alternating projection algorithm was proposed in [5] which makes use of the following property of the Frobenius norm:

$$\begin{aligned} \min_{S \in \Omega} \|Y - AS\|^2 &= \min_{\mathbf{s}(1) \in \Omega} \|\mathbf{y}(1) - A\mathbf{s}(1)\|^2 + \dots \\ &\quad + \min_{\mathbf{s}(N) \in \Omega} \|\mathbf{y}(N) - A\mathbf{s}(N)\|^2 \end{aligned} \tag{8}$$

Hence, minimisation over the signal vectors can be carried out independently. For each $\mathbf{s}(n)$ the ML estimate $\hat{\mathbf{s}}(n)$ is obtained by enumerating over all k^n possible vectors $\mathbf{s}^{(j)} \in \Omega$, and choosing the one which minimises

$$\hat{\mathbf{s}}(n) = \arg \min_{\mathbf{s}^{(j)} \in \Omega} \|\mathbf{y}(n) - A\mathbf{s}^{(j)}\|^2, \; j = 1 \dots k^n. \tag{9}$$

Whilst this approach provides an optimal solution, it quickly becomes computationally intractable if the number of sources to be separated or the alphabet are even moderately large. The application of Fisher Information Decision Directed optimisation provides a sub-optimal solution, but it is computationally achievable. Figure 4 shows the error performance for a signal matrix $S \in \mathbb{R}^{8 \times 100}$ and a mixing matrix $A \in \mathbb{R}^{9 \times 8}$ and compares the Fisher Information approach against a rounding algorithm. Because of the large amount of data available the performance of the rounding algorithm is better here than for the vector case of section three. However, for $p = 2$ the Fisher algorithm still offers an error performance improvement of several dB over the rounding algorithm.

5. CONCLUSION

The optimal solution to a finite alphabet optimisation problem can be found through exhaustive search, but for problems of even quite moderate size this approach quickly becomes computationally intractable. A common sub-optimal solution is to perform an unconstrained optimisation and then simply round each element of the result to the nearest member of the known alphabet. This paper has shown that Fisher Information Decision Directed optimisation provides better error performance than the rounding approach with only a moderate increase in complexity. For a binary alphabet we have shown that simultaneosly optimising two bits at a time provides an improved error performance

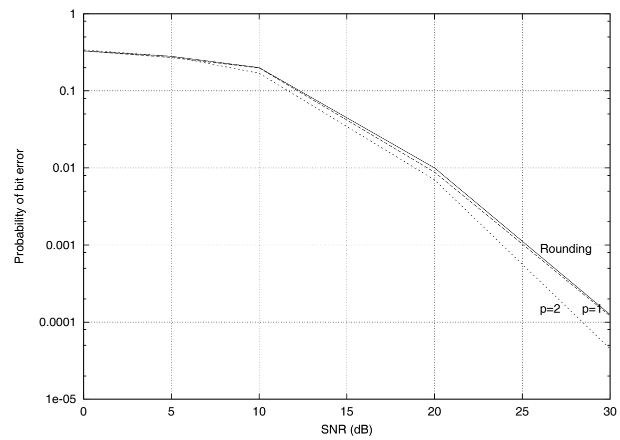

Fig. 4. Fisher Information Decision Directed optimisation of a BSS problem with eight sources

with no additional complexity over fixing one bit at a time. In addition, if we weight the probability of bit error by the increased number of iterations required then it has been shown that relative performance improves in a higher signal to noise ratio environment. It has been demonstrated that Fisher Information Decision Directed optimisation can be applied to the finite alphabet blind source separation problem when the number of sources makes an exhaustive search approach impractical.

6. REFERENCES

[1] Steven M. Kay. *Fundamentals of Statistical Signal Processing*, volume 1 of *Signal Processing*. Prentice-Hall, New Jersey, 1993.

[2] Qingyu Li, Er-Wei Bai, and Zhi Ding. Blind source separation of signals with known alphabets using ϵ-approximation algorithms. *IEEE Transactions on Signal Processing*, 51(1):1–10, Jan 2003.

[3] J. H. Manton, V. Ba-Ngu, and H. Ali. Fisher information decision directed quantisation: A fast sub-optimal solution to discrete optimisation problems with applications in wireless communications. In *International Conference on Optimization: Techniques and Applications*, Dec 2001.

[4] L.L. Scharf and L. McWhorter. Geometry of the cramer-rao bound. *Signal Processing*, 31:301–311, 1993.

[5] S. Talwar, M. Viberg, and A. Paulraj. Blind separation of synchronous co-channel digital signals using an antenna array. part i. algorithms. *IEEE Transactions on Signal Processing*, 44(5):1184–1197, May 1996.

[6] Mahesh K. Varanasi. Decision feedback multiuser detection: A systematic approach. *IEEE Transactions on Information Theory*, 45(1), 1999.

[7] A.J. Viterbi. Very low rate convolutional codes for maximum theoretical performance of spread spectrum multiple-access channels. *IEEE Journal on Selected Areas of Communications*, 8(4):641–649, May 1990.