EFFICIENT STATISTICAL PRUNING FOR MAXIMUM LIKELIHOOD DECODING

RADHIKA GOWAIKAR BABAK HASSIBI

California Institute of Technology
Department of Electrical Engineering, MC 136-93
Pasadena, CA 91125, USA
{gowaikar,hassibi}@systems.caltech.edu

ABSTRACT

In many communications problems, maximum-likelihood (ML) decoding reduces to finding the closest (skewed) lattice point in N-dimensions to a given point $x \in \mathcal{C}^N$. In its full generality, this problem is known to be NP-complete and requires exponential complexity in N. Recently, the expected complexity of the sphere decoder, a particular algorithm that solves the ML problem exactly, has been computed where it is shown that over a wide range of rates, SNRs and dimensions the expected complexity is polynomial in N. In this paper, we propose an algorithm that, for large N, offers substantial computational savings over the sphere decoder, while maintaining performance arbitrarily close to ML. The method is based on statistically pruning the search space. Simulations are presented to show the algorithm's performance and the computational savings relative to the sphere decoder.

1. INTRODUCTION

Multiple antenna systems have been shown to be capable of achieving high data rates. However, reliable decoding in these systems requires very high complexity. For a wide class of space-time transmission schemes (see e.g., [1, 2]) ML decoding requires us to solve an Integer Least Squares problem, which is, in general, NP-hard. Practical methods to solve this employ approximations or heuristics. Zero forcing cancellation, nulling and cancelling and nulling and cancelling with optimal ordering [1, 2] are some of these. However, the bit error rate (BER) performance of these is inferior to that of the exact methods.

Exact methods that search over the entire (finite) signal-space require exponential search. More sophisticated exact methods such as Kannan's algorithm [3], the KZ algorithm [4] and the sphere decoding algorithm of [5] attempt to reduce the search space. The branch and bound algorithm, popularly used to solve integer (usually linear) programming problems, imposes additional constraints on the optimizing variables to reduce the size of the problem and also requires estimates of upper and lower bounds of the objective function to prune the tree. Hence it is not suitable for the ML decoding problem.

In the sphere decoding algorithm we find the lattice points lying in a hypersphere centered at x and then determine the point closest to x. The analysis for the expected complexity of this has been done in [6]. While this algorithm yields polynomial-time complexity over a wide-range of rates, dimensions and SNRs, it does require a non-polynomial complexity for large N.

In this paper, we propose a modification to the sphere decoding algorithm that uses statistical pruning to reduce the search for the closest point to a region much smaller than the hypersphere. This causes a reduction in complexity, at the price of increasing the BER. However, we show that significant computational savings can be obtained while keeping the BER arbitrarily close to that of the ML decoder.

Below, we describe the system model and the original and modified decoding algorithms and then analyze the performance and complexity of the proposed algorithm.

2. SYSTEM MODEL

We assume a discrete-time block-fading multiple antenna channel model with N transmit and N receive antennas, where the channel is known to the receiver.[1] If $\mathcal{S}$ is the signal space, during any channel use the transmitted signal $\tilde{s} \in \mathcal{S}^{N\times 1}$ and the received signal $x \in \mathcal{C}^{N\times 1}$ are related by

$$x = \sigma_h H\tilde{s} + v \qquad (1)$$

where $H \in \mathcal{C}^{N\times N}$ is the known channel matrix and $v \in \mathcal{C}^{N\times 1}$ is the additive noise vector, comprised of independent, identically distributed (i.i.d.) complex-Gaussian entries $\mathcal{CN}(0,1)$ i.e. $\sigma_v^2 = 1$. If we assume that the entries

This work was supported in part by the National Science Foundation under grant no. CCR-0133818, by the office of Naval Research under grant no. N00014-02-1-0578, and by Caltech's Lee Center for Advanced Networking.

[1] The case of nonequal number of transmit/receive antennas can also be dealt with in a straightforward fashion, but is omitted for brevity.

of s and H have unit variance, then $\sigma_h = \sqrt{\frac{\rho}{N}}$ where ρ is the expected signal-to-noise ratio (SNR). Under the aforementioned assumptions the ML criterion requires us to find $s \in \mathcal{S}^{N\times 1}$ that minimizes $\|x - Hs\|^2$.

3. SPHERE DECODER

In sphere decoding we search only over lattice points that lie in a hypersphere of radius r around x, thus reducing the search space and the computation. Therefore we need to find all $s \in \mathcal{S}^{N\times 1}$ that satisfy

$$r^2 \geq \|x - Hs\|^2 \tag{2}$$

To this end, consider the QR decomposition of the channel matrix, $H = QR$ where R is an $N \times N$ upper triangular matrix with positive diagonal and Q is an $N \times N$ unitary matrix. We then have

$$\|x - Hs\|^2 = \|x - QRs\|^2 = \|Q^*x - Rs\|^2$$

Define $y' = Q^*x - Rs$ and $\lambda_i = |y'_{N-i+1}|^2$ for $i = 1, 2, \ldots N$. Note that, due to the upper-triangularity of R, λ_i depends only on the unknowns $s_N, \ldots, s_{N-i+1}$. Thus finding all s that satisfy (2) amounts to finding all s that satisfy

$$\lambda_1 + \lambda_2 + \ldots + \lambda_N \leq r^2$$

This is achieved by solving successively for

$$\begin{aligned} \lambda_1 &\leq r^2 \\ \lambda_1 + \lambda_2 &\leq r^2 \\ &\vdots \\ \lambda_1 + \lambda_2 + \cdots + \lambda_N &\leq r^2 \end{aligned} \tag{3}$$

The point is that the first condition gives an interval for s_N, whereas for any pre-determined $s_N, \ldots, s_{N-i+2}$, the i-th condition gives an interval for s_{N-i+1}.

We see that the algorithm constructs a search tree where the branches in the k-th level of the tree correspond to the lattice points inside the hypersphere of radius r and dimension k. The complexity of the algorithm depends on the *size* of the resulting search tree.

The radius r has to be chosen carefully. If r is too large, we obtain too many points, but if it is too small, we get no points in the hypersphere and have to redo the computation with a larger r. In [6], a choice of r based on the statistics of the noise is suggested. This r is proportional to N.

While the sphere decoding algorithm is one of the more efficient exact methods to solve the maximum likelihood problem with finite constellations (L-PAM, L-QAM etc.), it stops giving polynomial complexity at some N which is in the range of practical interest. The reason for this is understood as follows – because the radius r is proportional to N, the algorithm retains a very large fraction of the lattice points (in fact nearly all the points) upto some dimension k before it starts to prune the tree. For instance, if $N = 1000$, we have $r = \alpha N$ such that upto dimension $k = 100$ we keep nearly all the points of the lattice. This already gives us L^{100} points to search over and the complexity quickly becomes exponential.

4. STATISTICAL PRUNING

Taking our cue from this, we modify the algorithm to start pruning the tree corresponding to the search region much earlier. The sphere decoder gives exponential complexity for large N because the first several conditions of (3) are very loose and do not help in reducing the search space. We propose instead a schedule of radii $r_1 \leq r_2 \leq \ldots \leq r_N$:

$$\begin{aligned} \lambda_1 &\leq r_1^2 \\ \lambda_1 + \lambda_2 &\leq r_2^2 \\ &\vdots \\ \lambda_1 + \lambda_2 + \cdots + \lambda_N &\leq r_N^2 \end{aligned} \tag{4}$$

Denote by $\mathcal{D}_k$ the region in $\mathcal{S}^{k\times 1}$ containing points that satisfy the first k inequalities of (4). (Note that these points have been determined by finding out values of s_N, s_{N-1} $\ldots$, s_{N-k+1} that satisfy the first k conditions.) We refer to $\mathcal{D}_N$ as $\mathcal{D}$ in the following discussion. We can determine all $s \in \mathcal{D}$ by a procedure identical to that of the original sphere decoder. The algorithm is

Input: $Q, R, x, y = Q^*x, r_1, \ldots, r_N$.

1. Set $k = N, r'^2_N = r_1^2, y''_N = y_N$
2. Set $UB(s_k) = \lfloor \frac{r'_k + y''_k}{r_{k,k}} \rfloor$, $LB(s_k) = \lceil \frac{-r'_k + y''_k}{r_{k,k}} \rceil - 1$
3. $s_k = s_k + 1$. If $s_k \leq UB(s_k)$ go to 5, else go to 4.
4. $k = k + 1$ and go to 3.
5. $k = k - 1$. If $k = 0$, go to 6.
   Else
   $r'^2_k = r'^2_{k+1} + (r^2_{N-k+1} - r^2_{N-k}) - (y''_{k+1} - r_{k+1,k+1}s_{k+1})^2$
   $y''_k = y_k - \sum_{j=k+1}^{N} r_{k,j}s_j$
   Go to 2.
6. Solution found. Save s and go to 3.

Once all $s \in \mathcal{D}$ have been determined, we declare the decoder output as the $s \in \mathcal{D}$ which minimizes $\|x - Hs\|^2$.

Note that the region $\mathcal{D}$ is different from the hypersphere. Depending on the values of $r_1, r_2, \ldots r_N$ it may include more or less points than the hypersphere of radius r. To reduce the complexity, we naturally try to reduce the number of points in $\mathcal{D}$. However, because of the 'asymmetry' of

$\mathcal{D}$, it is possible that the lattice point closest to x does not lie in $\mathcal{D}$. Thus, unlike the sphere deoder, we are not doing ML decoding and are, potentially, incurring a greater BER.

Thus we obtain a tradeoff.

5. PROBABILITY OF ERROR

Let $\tilde{s}$ be the transmitted point and $\epsilon = P(\tilde{s} \notin \mathcal{D})$. With probability P_e we make an error by decoding to $s \neq \tilde{s}$.

$$\begin{aligned} P_e &= P_e(|\tilde{s} \in \mathcal{D})P(\tilde{s} \in \mathcal{D}) + P_e(|\tilde{s} \notin \mathcal{D})P(\tilde{s} \notin \mathcal{D}) \\ &\leq P_e(|\tilde{s} \in \mathcal{D})P(\tilde{s} \in \mathcal{D}) + 1 \cdot \epsilon \\ &= P(\|x - Hs\|^2 \leq \|v\|^2 \text{ for } s \neq \tilde{s}, s \in \mathcal{D} | \tilde{s} \in \mathcal{D}) \cdot \\ &\quad P(\tilde{s} \in \mathcal{D}) + \epsilon \\ &= P(\|x - Hs\|^2 \leq \|v\|^2 \text{ for } s \neq \tilde{s}, s \in \mathcal{D}, \tilde{s} \in \mathcal{D}) + \epsilon \\ &\leq P(\|x - Hs\|^2 \leq \|v\|^2 \text{ for } s \neq \tilde{s}) + \epsilon \\ &= P_e^{ML} + \epsilon \end{aligned}$$

where P_e^{ML} is the probability of error with ML decoding. The first inequality above is very loose and hence this is not a very tight upper bound. Also, since we are not using any coding on the transmitted signal, P_e^{ML} will not go to zero and so by making ϵ small we can obtain performance arbitrarily close to ML.

We now determine ϵ. If $s = \tilde{s}$, we have $y' = Q^*v$. Since Q is unitary, Q^*v has the same statistics as v i.e. i.i.d. entries distributed as $\mathcal{CN}(0,1)$. With $\lambda_i = |y'_{N-i+1}|^2$, we have $p_{\lambda_i}(\lambda_i) = e^{-\lambda_i}$. Because $\lambda_1, \ldots, \lambda_N$ are independent,

$$p_{\lambda_1,\lambda_2,\ldots,\lambda_N}(\lambda_1, \lambda_2, \ldots, \lambda_N) = e^{-(\lambda_1+\lambda_2+\cdots+\lambda_N)}$$

$1 - \epsilon$ is the probability that these λ_i's satisfy (4). Therefore

$$1 - \epsilon = \int_0^{r_1^2} \cdots \int_0^{r_N^2 - (\lambda_1+\cdots+\lambda_{N-1})} e^{-(\lambda_1+\cdots+\lambda_N)} d\lambda_N \ldots d\lambda_1$$

Changing variables to $\mu_i = \sum_{j=1}^{i} \lambda_j$ for $i = 1, \ldots, N$

$$1 - \epsilon = \int_0^{r_1^2} \int_{\mu_1}^{r_2^2} \cdots \int_{\mu_{N-1}}^{r_N^2} e^{-\mu_N} d\mu_N \ldots d\mu_1$$

If we call this integral I_N and integrate out μ_N we get

$$I_N = I_{N-1} - e^{-r_N^2} J_{N-1} \tag{5}$$

where $J_{N-1} = \int_0^{r_1^2} \int_{\mu_1}^{r_2^2} \cdots \int_{\mu_{N-2}}^{r_{N-1}^2} d\mu_{N-1} \ldots d\mu_1$. It can be shown that the J_i's satisfy the recurrence

$$J_k = \sum_{l=0}^{k-1} (-1)^{k-l+1} \frac{r_{l+1}^{2(k-l)}}{(k-l)!} J_l \tag{6}$$

We define $I_0 = 1$. Then, using (5) recursively, we get $I_N = 1 - \sum_{k=1}^{N} e^{-r_k^2} J_{k-1}$. The J_i's are determined by defining $J_0 = 1$ and using (6) recursively. We thus have

$$\epsilon = \sum_{k=1}^{N} e^{-r_k^2} J_{k-1} \tag{7}$$

We use this to determine the radii $r_1, \ldots, r_N$. For example, if we choose a linear schedule i.e. $r_i^2 = (\delta \log N + i)\sigma_v^2$, we choose δ such that $\epsilon = 0.01$ etc.

6. COMPUTATIONAL COMPLEXITY

To compute the complexity of the algorithm, we need to calculate the number of points that we search over. This means we need to determine how many points in $\mathcal{S}^{k\times 1}$ are also in $\mathcal{D}_k$ at every dimension $k = 1, \ldots, N$. We then need to sum over all dimensions to estimate the number of points visited during the decoding.

Let $s^k \in \mathcal{S}^{k\times 1}$. $s^k \in \mathcal{D}_k$ if it satisfies the first k equations of (4). Once $P(s^k \in \mathcal{D}_k)$ is determined, we can then sum these probabilites for all $s^k \in \mathcal{S}^{k\times 1}$ to get the expected number of points in the search space at dimension k. (Note here that at dimension k we have determined the values of $s_N, \ldots, s_{N-k+1}$ that satisfy the first k equations of (4).)

For any s^k, the joint distribution of $\lambda_1, \ldots, \lambda_k$ determines $P(s^k \in \mathcal{D}_k)$. More specifically,

$$\begin{aligned} &P(s^k \in \mathcal{D}_k) \\ &= \int_0^{r_1^2} \cdots \int_0^{r_k^2 - (\lambda_1+\cdots+\lambda_{k-1})} p_{\lambda_1,\ldots,\lambda_k}(\lambda_1, \ldots, \lambda_k) d\lambda_k \ldots d\lambda_1 \end{aligned} \tag{8}$$

If σ_v^2 is the variance of each entry of v ($\sigma_v = 1$), σ_h is as defined in section (2) and $c_i \triangleq \frac{1}{\sigma_v^2 + \sigma_h^2 \|\tilde{s}^i - \tilde{\tilde{s}}^i\|^2}$ ($\tilde{\tilde{s}}^i$ is the truncation of $\tilde{s}$ corresponding to $\mathcal{S}^{i\times 1}$ i.e. the vector $[\tilde{s}_N, \ldots, \tilde{s}_{N-i+1}]$) it can be shown that

$$p_{\lambda_i}(\lambda_i) = \frac{c_i^i}{c_{i-1}^{i-1}} e^{-c_i \lambda_i} \sum_{k=0}^{i-1} \binom{i-1}{k} \frac{\lambda_i^k}{k!} (c_{i-1} - c_i)^k \tag{9}$$

Since the λ_i's are independent we have

$$p_{\lambda_1,\ldots,\lambda_k}(\lambda_1, \ldots, \lambda_k) = \prod_{i=1}^{k} p_{\lambda_i}(\lambda_i) \tag{10}$$

Substituting (9) and (10) into (8), the integral for $P(s^k \in \mathcal{D}_k)$ can be obtained exactly. However, it gives an expression that is difficult to manipulate and sum over. Using some approximate analysis, it can be shown that

$$\begin{aligned} &\int_0^{r_1^2} \cdots \int_0^{r_k^2 - (\lambda_1+\cdots+\lambda_{k-1})} p_{\lambda_1,\ldots,\lambda_k}(\lambda_1, \ldots, \lambda_k) d\lambda_k \ldots d\lambda_1 \\ &\qquad \approx \prod_{i=1}^{k} \int_0^{X_i} p_{\lambda_i}(\lambda_i) \end{aligned}$$

where $X_1 \triangleq r_1^2$ and the X_i's are obtained by solving the recursion

$$\begin{aligned} X_i &= r_i^2 - r_{i-1}^2 - \frac{1 + \log 2}{2c_{i-1}} + \frac{X_{i-1}}{2} \\ &\quad + \frac{1}{2c_{i-1}} \sqrt{(1 + \log 2 + c_{i-1} X_{i-1})^2 - 4c_{i-1} X_{i-1}} \end{aligned}$$

It can further be proved that $\int_0^{X_i} p_{\lambda_i}(\lambda_i)$ is well approximated by $\min(1, \frac{c_i X_i}{2})$. We have omitted the details here in the interests of space.

We thus have

$$P(s^k \in \mathcal{D}_k) \approx \prod_1^k \min(1, \frac{c_i X_i}{2}) \qquad (11)$$

The complexity is now given by

$$\begin{aligned} C &= \sum_{k=1}^{m} (\text{expected \# of points in } \mathcal{D}_k) \cdot \underbrace{(\text{flops/point})}_{8k+32} \\ &= \sum_{k=1}^{N} (8k+32) \sum_{s^k \in \mathcal{S}^{k\times 1}} P(s^k \in \mathcal{D}_k) \\ &= \sum_{k=1}^{N} (8k+32) \sum_{s^k \in \mathcal{S}^{k\times 1}} \prod_1^k \min(1, \frac{c_i X_i}{2}) \end{aligned}$$

The above can be computed efficiently with Monte Carlo simulations. An exact sum also seems possible and we are working towards it.

7. SIMULATIONS

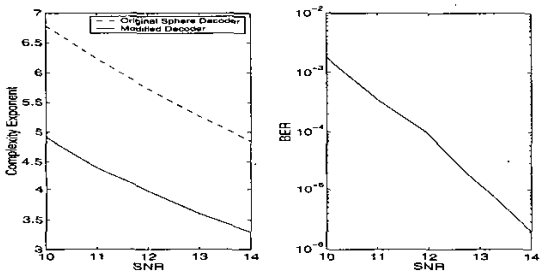

Fig. 1. Complexity Exponent and BER for N=50 with QPSK

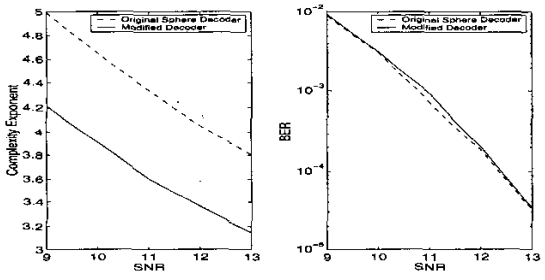

Fig. 2. Complexity Exponent and BER for N=20 with QPSK

Fig. (1) shows the BER and the complexity exponent i.e. $\log C / \log N$ for the modified decoder. We have N=50 and QPSK signalling. Since the constellation is complex, it amounts to decoding a $2N$=100-dimensional real signal. We have used $r_i^2 = (\delta \log N + i)\sigma_v^2$ with δ chosen to make $\epsilon = 0.9$. Fig. (1) also shows the complexity for the original sphere decoder. We can see that it requires nearly 50^2 times as much computation as the modified decoder. It is extremely computationally expensive to generate a BER plot with this decoder for N=50.

In order to compare the BER, we show results for N=20 and QPSK signalling in Fig. (2). We see that the loss in BER is quite insignificant and can be compensated by an increase in SNR of around 0.1dB. From the complexity exponent we see that the modified decoder runs around $20^{0.8}$ times faster.

8. CONCLUSIONS

We have an algorithm that performs nearly as well as ML decoding and gives significant savings in the computational complexity. With the modified sphere decoder sub-cubic complexities are possible for larger values of N in wider ranges of SNR than before. This is with BERs arbitrarily close to those for ML decoding. With different schedules the tradeoff between performance and complexity can be altered.

It will be interesting to see how this generalizes to systems that include coding in the signalling scheme.

9. REFERENCES

[1] G. J. Foschini, "Layered space-time architecture for wireless communication in a fading environment when using multi-element antennas," *Bell Labs. Tech. J.*, vol. 1, no. 2, pp. 41–59, 1996.

[2] B. Hassibi and B. Hochwald, "High-rate codes that are linear in space and time," *IEEE Trans. Info. Theory*, vol. 48, no. 7, pp. 1804–1824, July 2002.

[3] R. Kannan, "Improved algorithms on integer programming and related lattice problems," *Proc. 15th Annu. ACM Symp. on Theory of Computing*, pp. 193–206, 1983.

[4] J.C. Lagarias, H.W. Lenstra, and C.P. Schnorr, "Korkin-Zolotarev bases and successive minima of a lattice and its reciprocal," *Combinatorica*, vol. 10, pp. 333–348, 1990.

[5] U. Fincke and M. Pohst, "Improved methods for calculating vectors of short length in a lattice, including a complexity analysis," *Mathematics of Computation*, vol. 44, no. 170, pp. 463–471, April 1985.

[6] B. Hassibi and H. Vikalo, "On the expected complexity of sphere decoding," *Submitted to the IEEE Transactions on Signal Processing*, 2002.