ON THE EXPECTED COMPLEXITY OF SPHERE DECODING

Babak Hassibi

Department of Electrical Engineering
California Institute of Technology
Pasadena, CA 91125

Haris Vikalo

Information Systems Laboratory
Stanford University
Stanford, CA 94305

ABSTRACT

The problem of finding the least-squares solution to a system of linear equations where the unknown vector is comprised of integers, but the matrix coefficient and given vector are comprised of real numbers, arises in many applications: communications, cryptography, GPS, to name a few. The problem is equivalent to finding the closest lattice point to a given point and is known to be NP-hard. In communications applications, however, the given vector is not arbitrary, but rather is an unknown lattice point that has been perturbed by an additive noise vector whose statistical properties are known. Therefore in this paper, rather than dwell on the worst-case complexity of the integer-least-squares problem, we study its expected complexity, averaged over the noise and over the lattice. For the "sphere decoding" algorithm of Fincke and Pohst we find a closed-form expression for the expected complexity and show that for a wide range of noise variances the expected complexity is polynomial, in fact often sub-cubic. Since many communications systems operate at noise levels for which the expected complexity turns out to be polynomial, this suggests that maximum-likelihood decoding, which was hitherto thought to be computationally intractable, can in fact be implemented in real-time—a result with many practical implications.

1. THE INTEGER LEAST-SQUARES PROBLEM

In this paper we shall be concerned with the following so-called *integer least-squares problem*

$$\min_{s\in\mathcal{Z}^m} \|x - Hs\|_2 , \qquad (1)$$

where $x \in \mathcal{R}^n$, $H \in \mathcal{R}^{n\times m}$, and $\mathcal{Z}^m$ denotes the m-dimensional integer lattice. Note that the search space $\mathcal{Z}^m$ is discrete, yet infinite, and that the difficulty in solving (1) stems from the discreteness of the search space. Often, the search space is a (finite) subset of a lattice, $\mathcal{D} \subset \mathcal{Z}^m$, in which case we have

$$\min_{s\in\mathcal{D}\subset\mathcal{Z}^m} \|x - Hs\|_2 . \qquad (2)$$

Problems (1) and (2) have the following interpretations: given the "skewed" lattice Hs, or a finite subset thereof, find the "closest" lattice point to a given n-dimensional vector x. Integer least-squares problems arise in many communications problems (see, e.g., [1] for an application in CDMA systems and [2] for one in space-time codes), as well as in global positioning systems (GPS) [3]. Problems (1) and (2) are well known to be NP-hard, both in a worst-case and in an average sense [4]. In fact, there is a whole family of public-key cryptosystems based on the NP-hardness of the integer least-squares problem [5, 6].

1.1. Heuristic Methods

All practical systems therefore resort to approximations and/or heuristics to solve (1)-(2). Some common heuristics include the following:

1. Solve the unconstrained least-squares problem and then round-off to the closest integer. This is called zero-forcing equalization or Babai estimation [7].

2. *Nulling and cancelling:* Use only the Babai estimate for one of the entries of s, say s_1. Assume then that s_1 is known and subtract out its effect to obtain a reduced-order integer least-squares problem with $m-1$ unknowns. Solve similarly for s_2, etc. In communications parlance, this method is known as *decision-feedback equalization.*

3. *Nulling and cancelling with optimal ordering:* Perform nulling/cancelling, ordered from the "strongest" to the "weakest" signal (see [8, 9]).

All the above heuristic solutions require $O(m^3)$ computations. They will always give the exact solution if the columns of H are orthogonal, but, unfortunately, this is rarely the case. Orthogonalizing the columns of H via a QR decomposition, or otherwise, generally destroys the lattice structure. Therefore one often uses a *lattice reduction* method: find an invertible matrix T, such that T and T^{-1} have integer entries, and such that the matrix $G = HT$ is as "orthogonal as

0-7803-7147-X/01/$10.00©2001 IEEE

possible". In this case, instead of (1), we can solve

$$\min_{t \in \mathcal{Z}^m} \|x - Gt\|_2 ,$$

using the earlier heuristics and set $s = T^{-1}t$. A common algorithm for lattice reduction is the LLL (Lenstra, Lenstra and Lovasz) algorithm [7].

Although lattice reduction may be useful for integer least-squares problems over the infinite lattice (1), they are generally not useful when solving over subsets of a lattice (2) since they destroy the properties of the subset $\mathcal{D} \subset \mathcal{Z}^m$.

2. SPHERE DECODING

In addition to the aforementioned heuristic methods, there also exist exact methods that are a bit more sophisticated than performing a full search over the entire integer lattice. One is Kannan's algorithm [10], which searches only over lattice points in a carefully selected rectangular parallelepiped, and the other is the sphere decoding algorithm of Fincke and Pohst [11], which has recently been suggested for various communications problems [1, 12]. Nevertheless, both these algorithms require exponential complexity. In the remainder of this paper we shall focus on the sphere decoding algorithm.

The main idea in sphere decoding is to search over only lattice points that lie in a certain hypersphere of radius r around x, rather than to search over the entire integer lattice, thereby reducing the required computations. Clearly, the closest lattice point inside the hypersphere will also be the closest lattice point for the whole lattice. Although this seems to be a promising idea, there are two questions that come up.

1. *How to choose r?* Clearly, if r is too large, we obtain too many points, but if r is too small, we obtain no points. A natural candidate is the *covering radius*, defined to be the radius of the spheres centered at the lattice points that cover the whole space in the most economical way. Unfortunately, computing the covering radius is itself NP-hard.

2. *How can we tell which lattice points are inside the sphere?* If this requires testing each lattice point, then there is no point in sphere decoding.

The algorithm of Fincke and Pohst does not really address the first question. However, it does propose an efficient way to answer the second one. To this end, note that s lies in a sphere of radius r iff

$$r^2 \geq \|x - Hs\|^2 = (s-\hat{s})^* H^* H (s-\hat{s}) + \|x\|^2 - \|H\hat{s}\|^2 .$$

where $\hat{s} = H^{\dagger}x$. Introducing the QR decomposition $H = QR$, and defining $r'^2 = r^2 - \|x\|^2 + \|H\hat{s}\|^2$, we can write the above as

$$\begin{aligned} r'^2 &\geq (s-\hat{s})^* R^* R (s-\hat{s}) \\ &= \sum_{i=1}^{m} r_{ii}^2 \left(s_i - \hat{s}_i + \sum_{j=i+1}^{m} \frac{r_{ij}}{r_{ii}} (s_j - \hat{s}_j) \right)^2 \\ &= r_{mm}^2 (s_m - \hat{s}_m)^2 + \\ &\quad r_{m-1,m-1}^2 \left(s_{m-1} - \hat{s}_{m-1} + \frac{r_{m-1,m}}{r_{m-1,m-1}} (s_m - \hat{s}_m) \right)^2 + \cdots \end{aligned}$$

A necessary condition for s to lie inside the sphere is therefore that

$$r_{mm}^2 (s_m - \hat{s}_m)^2 \leq r'^2 .$$

But this condition is easy to check and leads to

$$\left\lceil \hat{s}_m - \frac{r'}{r_{mm}} \right\rceil \leq s_m \leq \left\lfloor \hat{s}_m + \frac{r'}{r_{mm}} \right\rfloor . \qquad (3)$$

This condition is by no means sufficient. For every s_m satisfying (3), defining $r'^2_{m-1} = r'^2 - r_{mm}^2 (s_m - \hat{s}_m)^2$, a stronger necessary condition is

$$r_{m-1,m-1}^2 \left(\underbrace{s_{m-1} - \hat{s}_{m-1} + \frac{r_{m-1,m}}{r_{m-1,m-1}} (s_m - \hat{s}_m)}_{\hat{s}_{m-1|m}} \right)^2 \leq r'^2_{m-1},$$

which is readily equivalent to

$$\left\lceil \hat{s}_{m-1|m} - \frac{r'_{m-1}}{r_{m-1,m-1}} \right\rceil \leq s_{m-1} \leq \left\lfloor \hat{s}_{m-1|m} + \frac{r'_{m-1}}{r_{m-1,m-1}} \right\rfloor .$$

One can continue in a similar fashion for s_{m-2}, and so on, until all points inside the sphere are found. This essentially leads us to the sphere decoding algorithm.

The Algorithm

Input: $R, x, \hat{s}, r$.

1. Set $k = m$, $r'^2_m = r^2 - \|x\|^2 + \|H\hat{s}\|^2$, $\hat{s}_{m|m+1} = \hat{s}_m$
2. (Bounds for s_k) Set $z = \frac{r'_k}{r_{kk}}$, $UB(s_k) = \lfloor z + \hat{s}_{k|k+1} \rfloor$, $s_k = \lceil -z + \hat{s}_{k|k+1} \rceil - 1$
3. (Increase s_k) $s_k = s_k + 1$. If $s_k \leq UB(s_k)$ go to 5, else to 4.
4. (Increase k) $k = k + 1$ and go to 3.
5. (Decrease k) If $k = 1$ go to 6. Else $k = k - 1$, $\hat{s}_{k|k-1} = \hat{s}_k + \sum_{j=k+1}^{m} \frac{r_{kj}}{r_{kk}} (s_j - \hat{s}_j)$, $r'^2_k = r'^2_{k+1} - r^2_{k+1,k+1} (s_{k+1} - \hat{s}_{k+1|k+2})^2$.
6. Solution found. Save s_k and go to 3.

Remark: Rather than search over all lattice points in a sphere of radius r and dimension m, the algorithm searches over all lattice points in spheres of radius r and dimensions $1, 2, \ldots, m$. The algorithm therefore constructs a tree, where the branches in the k-th level of the tree correspond to the lattice points inside the sphere of radius r and dimension k. This is depicted in Fig. 1.

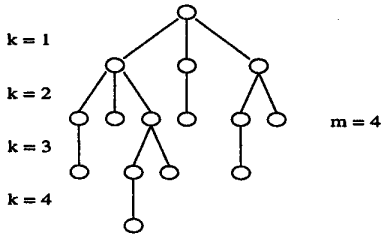

Fig. 1. Tree generated by sphere decoding algorithm.

2.1. A First Look at Complexity

A heuristic study of the expected complexity of the sphere decoding algorithm can be given as follows. For an arbitrary point x, the expected number of lattice points inside a k-dimensional sphere of radius r is proportional to the volume

$$\frac{\pi^{k/2}}{\Gamma(k/2+1)}r^k,$$

where $\Gamma(\cdot)$ is the Gamma function. Therefore the expected total number of points visited by the algorithm is proportional to

$$\sum_{k=1}^{m}\frac{\pi^{k/2}}{\Gamma(k/2+1)}r^k > \sum_{k=1}^{\frac{m}{2}}\frac{\pi^{k}}{\Gamma(k+1)}r^{2k} \approx e^{\pi r^2}, \text{ for large } m.$$

To have a nonvanishing probability of finding a point in the m-dimensional sphere, its volume must be $\frac{\pi^{m/2}}{(m/2)!}r^m = O(1)$. But from Stirling's formula this implies that $r^2 = O(m)$ and that the complexity of the algorithm is exponential, $e^{O(m)}$.

3. A RANDOM MODEL

Although not unexpected, the above complexity analysis yields a discouraging result. In communications applications, however, the vector x is not arbitrary, but rather is a lattice point perturbed by additive noise with known statistical properties. Thus, we will assume

$$x = Hs + v,$$

where the entries of v are independent $N(0,\sigma^2)$ random variables. It is also useful, and not unreasonable, to assume that the lattice-generating-matrix H is random and is comprised of independent $N(0,1)$ entries. We further assume, for simplicity, that $m = n$. (The more general case of $m \neq n$ can also be studied without much more effort.)

The first by-product of these assumptions is a method to determine the desired radius r. Note that $\|v\|^2 = \|x - Hs\|^2$ is a Ξ^2 random variable with $m/2$ degrees of freedom. Thus we may choose the radius $r^2 = \alpha m\sigma^2$ in such a way that we find a lattice point with high probability:

$$\int_0^{\alpha m}\frac{\lambda^{m/2-1}}{\Gamma(m/2)}e^{-\lambda}d\lambda = 0.99, \quad \text{say.}$$

The expected complexity can now be given by

$$\sum_{k=1}^{m}\underbrace{(\text{expected \# of points in } k\text{-sphere of radius } r)}_{\triangleq E_p(k,r^2=\alpha m\sigma^2)}\cdot\underbrace{(\text{flops/point})}_{2k+17}$$

The question is how to compute $E_p(k,r^2)$? Suppose that the lattice point s_t was transmitted and that the vector $x = Hs_t + v$ was received. The probability that an arbitrary lattice point s_a lies in a hypersphere of radius r around x can be computed to be

$$\gamma\left(\frac{r^2}{\sigma^2+\|s_a-s_t\|^2},\frac{k}{2}\right) = \int_0^{\frac{r^2}{\sigma^2+\|s_a-s_t\|^2}}\frac{\lambda^{k/2-1}}{\Gamma(k/2)}e^{-\lambda}d\lambda.$$

Note that the above probability depends only on $\|s_a - s_t\|^2 = \|s\|^2$, i.e., on the squared norm of an arbitrary lattice point in the k-dimensional lattice. It is thus straightforward to see that

$$E_p(k,r^2) = \sum_{n=0}^{\infty}\gamma\left(\frac{r^2}{\sigma^2+n},\frac{k}{2}\right)\cdot(\text{\# of lattice points with } \|s\|^2 = n).$$

Since $\|s\|^2 = s_1^2 + \ldots + s_k^2$, we basically, need to figure out how many ways a non-negative integer n can be represented as the sum of k squared integers. This is a classic problem in number theory and the solution is denoted by $r_k(n)$ [13]. There exist a plethora of results on how to compute $r_k(n)$, its asymptotic values (in k and n), etc. We only mention one here: $r_k(n)$ is given by the coefficient of x^n in the expansion

$$\left(1+2\sum_{m=1}^{\infty}x^{m^2}\right)^k = 1+\sum_{n=1}^{\infty}r_k(n)x^n.$$

The above arguments lead to the following result.

Theorem 1 (Expected complexity for problem (1)) *Under the aforementioned assumptions, the expected complexity of the sphere decoder for problem (1) is given by*

$$C(m,\sigma^2) = \sum_{k=1}^{m}(2k+17)\sum_{n=0}^{\infty}r_k(n)\gamma\left(\frac{\alpha m\sigma^2}{\sigma^2+n},\frac{k}{2}\right),$$

where α is such that $\gamma(\alpha m, m) = 1-\epsilon$.

It is often useful to look at the *complexity exponent*

$$\frac{\log C(m,\sigma^2)}{\log m},$$

which approaches a constant if the expected complexity is polynomial, and grows like $\frac{m}{\log m}$ if it is exponential. The

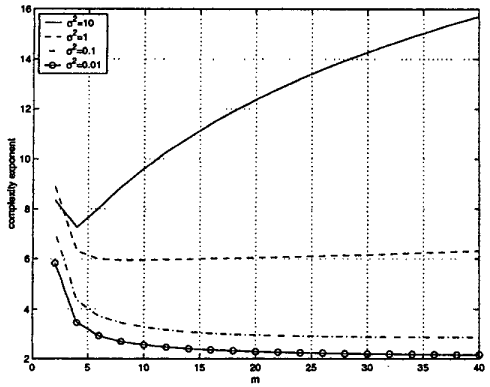

Fig. 2. *The complexity exponent as a function of m for $\sigma^2 = 0.01, 0.1, 1, 10$.*

complexity exponent is plotted as a function of m for different values of the σ^2 in Fig. 2. As can be seen, for small enough noise the expected complexity is polynomial, whereas for large noise it is exponential.

In communications problems, we are usually concerned with L-PAM constellations

$$\mathcal{D}_L^m = \left\{ -\frac{L-1}{2}, -\frac{L-3}{2}, \ldots, \frac{L-3}{2}, \frac{L-1}{2} \right\}^m .$$

In this case, rather than the noise variance σ^2, one is interested in the SNR, $\rho = \frac{m(L^2-1)}{12\sigma^2}$. For such constellations, computing the expected complexity is more involved than for the infinite lattice. Nonetheless, we have the following result.

Theorem 2 (Expected complexity for problem (2)) *Under the aforementioned assumptions, the expected complexity of the sphere decoder for problem (2) for a 2-PAM constellation is*

$$C(m,\rho) = \sum_{k=1}^{m} (2k+17) \sum_{n=0}^{k} \binom{k}{n} \gamma\left(\frac{\alpha m}{1 + \frac{12\rho n}{m(L^2-12)}}, \frac{k}{2} \right).$$

For a 4-PAM constellation it is

$$\sum_{k=1}^{m} (2k+17) \sum_{n} \frac{1}{2^k} \sum_{l=0}^{k} \binom{k}{l} g_{kl}(n) \gamma\left(\frac{\alpha m}{1 + \frac{12\rho n}{m(L^2-12)}}, \frac{k}{2} \right),$$

where $g_{kl}(n)$ is the coefficient of x^n in the polynomial

$$(1 + x + x^4 + x^9)^l (1 + 2x + x^4)^{k-l}.$$

Similar expressions can be obtained for 8-PAM, 16-PAM, etc., constellations.

Fig. 3 shows the complexity exponent as a function of m for $\rho = 20$db, for different L-PAM constellations with $L = 2, 4, 8, 16$. As can be seen, for low rates (i.e., small constellations) the expected complexity is polynomial, whereas for high rates (i.e., large constellations) the expected compexity is exponential. Simulation results suggest that, for a given SNR, the complexity is polynomial as long as the rate is less than the Shannon capacity corresponding to the SNR. This suggests that maximum-likelihood decoding can be feasible, since communication systems operate at rates below capacity.

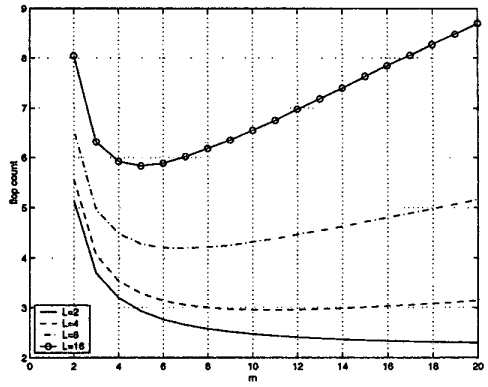

Fig. 3. *The complexity exponent as a function of m for $\rho = 20db$ and $L = 2, 4, 8, 16$.*

Finally, Fig. 4 shows the improvement in performance of sphere decoding over nulling and cancelling for a certain multi-antenna space-time code corresponding to $m = 64$ (for the details see [2]). The complexity of ML decoding via the sphere decoder is only twice the complexity of nulling and cancelling, whereas the performance improvement is significant. The number of points in the lattice subset for this example is 10^{38}. Thus, the sphere decoder yields the maximum-likelihood estimate in (roughly) cubic time, rather than having to search over all 10^{38} possibilities.

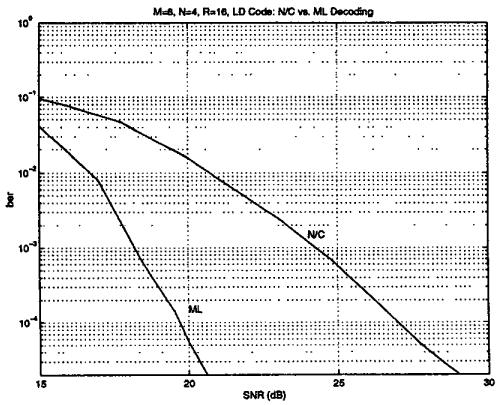

Fig. 4. *Sphere decoder (ML) vs. nulling and cancelling with optimal ordering (NC).*

4. CONCLUSION

In many communication problems, CDMA systems, signalling over FIR channels, space-time codes, etc., maximum-likelihood detection reduces to solving an integer least-squares problem. In such applications maximum-likelihood detection is rarely performed, on the grounds that it requires exponential complexity and is therefore computationally intractable. While it is true that the worst-case and expected complexity of solving the integer least-squares problem is exponential when the observed vector is arbitrary, in communications problems the observed vector is not arbitrary, but is rather a lattice point perturbed by additive noise whose statistics are known. In fact, communication systems operate at rates below the Shannon capacity, which typically implies that the noise variance is much less than the distance between the lattice points.

In this paper we studied the expected complexity of the integer-least-squares problem, averaged over the noise and over the lattice. For the sphere decoding algorithm of Fincke and Pohst we found a closed-form expression for the expected complexity in terms of the noise variance, the dimension of the lattice, and (for subsets of lattices) the constellation. It turns out that over a wide range of noise variances and dimensions the expected complexity is polynomial, in fact often cubic or sub-cubic. Since many communications systems operate at noise levels for which this is the case, this suggests that maximum-likelihood decoding, which was hitherto thought to be computationally intractable, can in fact be implemented with complexity similar to heuristic methods, but with siginificant performance gains—a result with many practical implications.

5. REFERENCES

[1] C. Brutel and J. Boutros, "Euclidean space lattice decoding for joint detection in CDMA systems," in *Proc. of the 1999 IEEE Information Theory and Communications Workshop*, p. 129, 1999.

[2] B. Hassibi and B. Hochwald, "High-rate codes that are linear in space and time," *submitted to IEEE Trans. Info. Theory*, 2000. Download available at `http://mars.bell-labs.com`.

[3] A. Hassibi and S. Boyd, "Integer parameter estimation in linear models with applications to GPS," *IEEE Transactions on Signal Processing*, vol. 46, pp. 2938–52, November 1998.

[4] M. Ajtai, "The shortest vector problem in L_2 is NP-hard for randomized reductions," in *Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, pp. 10–19, 1998.

[5] M. Ajtai, "Generating hard instances of lattice problems," in *Proceedings of the 28th Annual ACM Symposium on Theory of Computing*, pp. 99–108, 1996.

[6] O. Goldreich, S. Goldwasser, and S. Halevi, "Public-key cryptosystems from lattice reduction problems," in *Advances in Cryptology - CRYPTO97. 17th Annual International Cryptology Conference*, pp. 112–31, 1997.

[7] M. Grotschel, L. Lovasz, and A. Schrijver, *Geometrical Algorithms and Combinatorial Optimization*. New York, NY: Springer-Verlag, 1993.

[8] 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.

[9] B. Hassibi, "An efficient square-root algorithm for BLAST," *submitted to IEEE Trans. Sig. Proc.*, 1999. Download available at `http://mars.bell-labs.com`.

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

[11] 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, pp. 463–471, April 1985.

[12] M. O. Damen, A. Chkeif, and J.-C. Belfiore, "Lattice code decoder for space-time codes," *IEEE Comm. Let.*, pp. 161–163, May 2000.

[13] G. Hardy, *Ramanujan: Twelve Lectures*. Chelsea Publishing, 1940.