# Randomised Algorithms for Integer Signal Recovery

Student: Yingzi Xu  Supervisor: Prof. Xiao-Wen Chang

McGill University

## Abstract

Integer least squares problem is crucial in the field of communication and has wide applications. In this project we consider the randomised algorithm proposed by Klein [1], which is based on Babai point proposed in [2]. We studied the performance of this randomised algorithm as well as the effect of parameters. We also showed a lower bound for the success probability of Klein's algorithm, and we proved the algorithm samples Babai point with probability 1 by varying a parameter.

## Introduction

The problem is given as follows. Given a model matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$ of full column rank, a measurement vector $\mathbf{y}$, the linear model is defined as $\mathbf{y} = \mathbf{Ax} + \mathbf{v}$, where $\mathbf{x} \in \mathbb{Z}^n$ is an unknown parameter vector. The goal is to solve the following problem:

$$\min_{\mathbf{x}\in\mathbb{Z}^n} ||\mathbf{y} - \mathbf{Ax}||_2^2, \quad \mathbf{v} \sim N(\mathbf{0}, \sigma^2\mathbf{I})$$

Babai point is deterministic, fast to compute and usually performs well as a solution to the above problem when noise is small. Klein's algorithm is a randomised version of Babai point. It builds a multivariate Gaussian-like distribution and samples a lattice point. Given input vector $\mathbf{y}$, the probability of sampling a lattice point $\hat{\mathbf{y}}$ is lower bounded by

$$\frac{\exp(-G||\mathbf{y} - \hat{\mathbf{y}}||^2)}{\prod_{i=1}^n s(Gr_{ii}^2)} \tag{1}$$

where $G$ is a parameter,
$s(c) = \sum_{x\geq 0} e^{-cx^2} + e^{-c(1+x)^2}$ is a function, $r_{ij}$ is the $(i, j)$-th entry of the upper triangular matrix $\mathbf{R}$ from QR decomposition of $\mathbf{A}$.

## Lower Bound for Success Probability

The lower bound shown in (1) is a conditional probability, and by integration over all possible $\mathbf{v} \in \mathbb{R}^n$ we have the success probability lower bound for Klein's algorithm as

$$\prod_{i=1}^n \frac{1}{s(Gr_{ii}^2)}(1 + 2\sigma^2 G)^{-\frac{n}{2}} \tag{2}$$

which no longer depends on the input measurement vector $\mathbf{y}$.

## Effect of Parameter G

We find a proof for the following lemma:
*As $G \to +\infty$, Klein's randomised algorithm samples Babai point with probability 1.*
In other words, the parameter $G$ controls the density of the discrete multivariate Gaussian-like distribution. The greater $G$ is, the more dense the distribution is around Babai point.
Please refer to the project report for detailed proof.

## Maximising Lower Bound

By maximising the probability lower bound, we can find the optimal choice of parameter $G$ which maximises the success probability of the algorithm.
The optimal $G$ can be obtained by finding the root of the function

$$g_{\mathbf{A},\sigma^2}(G) = \sum_{i=1}^n \frac{2r_{ii}^2 \sum_{j\geq 1} j^2 e^{-Gr_{ii}^2 j^2}}{1 + 2\sum_{j\geq 1} e^{-Gr_{ii}^2 j^2}} - \frac{n\sigma^2}{1 + 2\sigma^2 G} \tag{3}$$

## Sampling Multiple Times

A natural idea to apply is that we run Klein's algorithm $k$ times, and we take the "best" lattice point as our final output. This will take more time but from our experiments the success rate can be significantly improved.
When $k \geq 5$, the performance of the randomised algorithm exceeds that of Babai point.

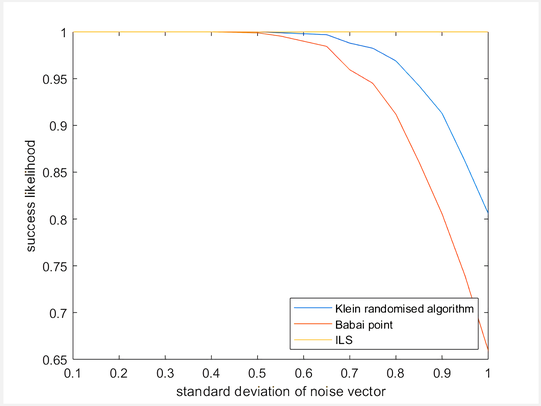


Figure: Success likelihood plot, with $\mathbf{A} \in \mathbb{R}^{100\times 80}$, k=10. Best point in terms of Euclidean distance from $\mathbf{y}$.

Also this algorithm's running time does not depend on the standard deviation of noise vector, therefore when noise is large, the computation time of Klein's algorithm is shorter compared to sphere decoding (labelled as "ILS" in plot).
Therefore we can regard Klein's algorithm as a in-between solution between Babai point and sphere decoding when noise is large, since it takes relatively less time than sphere decoding, and success rate is higher than Babai point.

## Future Work (In Progress)

- Finding tighter lower bounds for the success probability
- Success probability derivation of the closest point among $k$ sample points from the randomised algorithm
- New parameters in the algorithm to improve the probability distribution
- The effect of LLL-reduction on the success probability improvement of the algorithm
- Considering box-constrained cases instead of general cases, which are more practical in real-life applications.
- Performance of the randomised algorithm with different signal-to-noise ratios $\frac{||\mathbf{Ax}^*||}{\sigma}$

## References

[1] Philip Klein.
Finding the closest lattice vector when it's unusually close.
In *Proceedings of the eleventh annual ACM-SIAM symposium on Discrete algorithms*, pages 937–941. Society for Industrial and Applied Mathematics, 2000.

[2] L. Babai.
On lovász' lattice reduction and the nearest lattice point problem.
*Combinatorica*, 6(1):1–13, Mar 1986.