Enumeration

Phong Nguyễn
http://www.di.ens.fr/~pnguyen

March 2017

References

- Joint work with:
  - Yoshinori Aono, published at EUROCRYPT 2017: « Random Sampling Revisited: Lattice Enumeration with Discrete Pruning ». Full version on eprint.
  - Nicolas Gama and Oded Regev, published at EUROCRYPT 2010: « Lattice Enumeration with Extreme Pruning ».

Schnorr's Random Sampling [Sc03]

- The records [KaTe,KaFu] used a secret variant of RSR.
- RSR is based on Random Sampling, which is not well-understood, and which we revisit.

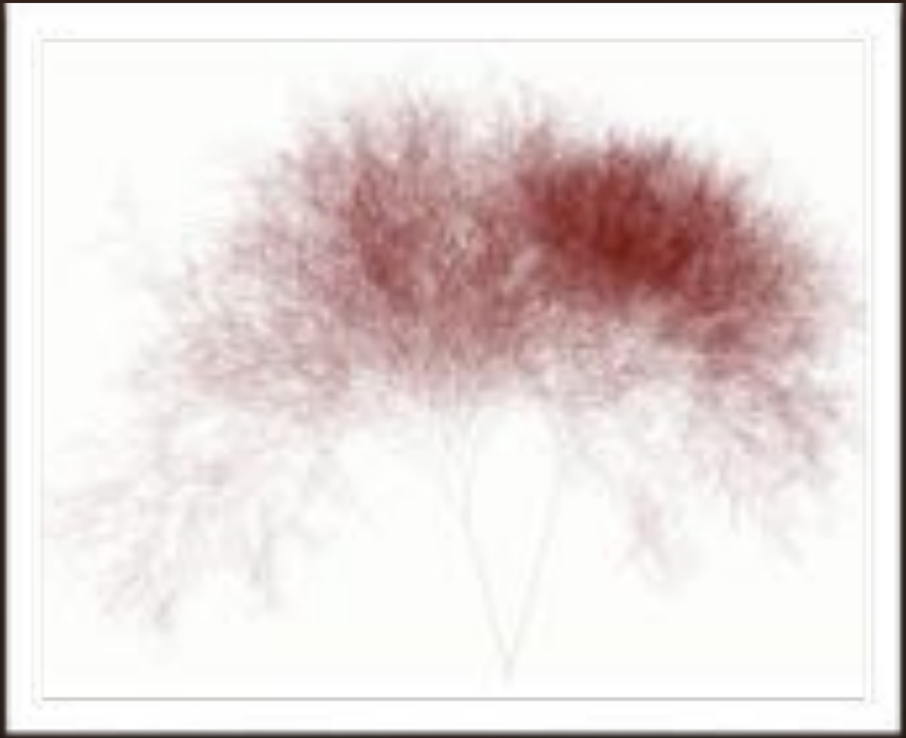
Revisiting and Unifying Schnorr's Algorithms

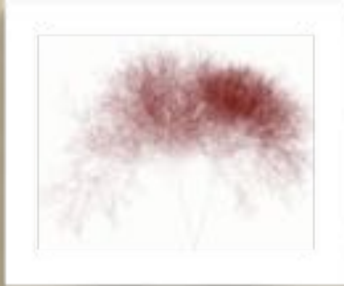
- Cylinder pruning
  - [SchnorrEuchner94,SchorrHorner95] but analysis not satisfactory;
  - Revisited in [GNR10]: better description led to better analysis, which led to much better performances.
- Random sampling [Schnorr03, BuLu06, FuKa15, etc.]
  - Previous analyses arguably not satisfactory: gap between analysis and experiments.
  - Discrete pruning [AoN17] generalizes it and provides a [GNR10]-type analysis.

Summary

- Enumeration
- Enumeration with Pruning
    - Cylinder Pruning
    - Discrete Pruning or Box Pruning

Solving SVP by Enumeration





Enumeration

- It is the simplest method to solve hard lattice problems: SVP, CVP, etc. Unrelated to bounds on Hermite's constant, but used in largest records.

- Input: a lattice L and a small ball $S \subseteq \mathbf{R}^n$ s.t. $\#(L \cap S)$ is « small ».

- Output: All points in $L \cap S$.

- Drawback: the running-time is typically superexponential, much larger than $\#L \cap S$.

Enumeration

- A) Reduce a basis.
- B) Exhaustive search all vectors ≤ R by enumerating all short vectors in projected lattices.

- Usually, B) is much more expensive than A).
- If the basis is only LLL-reduced, B) costs $2^{O(d^2)}$.
- [Kannan1983] showed that A) and B) can be done in $2^{O(d \ln d)}$ poly-time operations.

Enumeration

- Idea: projecting a vector can only shorten it.
- Enumeration is a depth-first search of a gigantic tree, to find a shortest vector.

The nb of tree nodes can be ``predicted" with the Gaussian heuristic [HaSt07,GNR10]

More precisely…

- Consider a lower-triangular matrix:

x_1	$b_{1,1}$				
x_2	$b_{2,1}$	$b_{2,2}$			
x_3	$b_{3,1}$	$b_{3,2}$	$b_{3,3}$		
x_4	$b_{4,1}$	$b_{4,2}$	$b_{4,3}$	$b_{4,4}$	
x_5	$b_{5,1}$	$b_{5,2}$	$b_{5,3}$	$b_{5,4}$	$b_{5,5}$

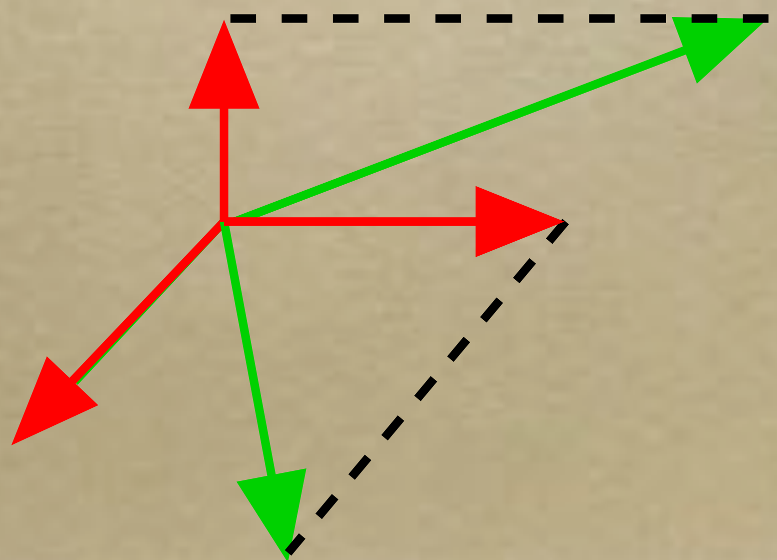
- If norm ≤ R, then
  - $(x_5 b_{5,5})^2 \leq R^2$
  - $(x_4 b_{4,4} + x_5 b_{5,4})^2 + (x_5 b_{5,5})^2 \leq R^2$
  - ...
  - So enumerate x_5, then x_4, etc.

Remember Gram-Schmidt

- From d linearly independent vectors, GS constructs d orthogonal vectors: the i-th vector is projected over the orthogonal complement of the first i-1 vectors.

$$\vec{b}_1^\star = \vec{b}_1$$
$$\vec{b}_i^\star = \vec{b}_i - \sum_{j=1}^{i-1} \mu_{i,j} \vec{b}_j^\star$$

where $\mu_{i,j} = \frac{\langle \vec{b}_i, \vec{b}_j^\star \rangle}{\|\vec{b}_j^\star\|^2}$



Remember Projections

- Denote by π_i the projection orthogonally to $b_1,\ldots,b_{i-1}$.
- Then:
  - $b_i^* = \pi_i(b_i)$
  - $\pi_i(L)$ is a lattice of dim $d-i+1$ whose volume is $\mathrm{vol}(L)/(||b_1^*|| \times \ldots \times ||b_{d-i+1}^*||) = \mathrm{vol}(L)/\mathrm{vol}(b_1,\ldots,b_{i-1})$.

Gram-Schmidt = Triangularization

- If we take an appropriate orthonormal basis, the matrix of the lattice basis becomes triangular.

$$
\begin{pmatrix}
\|\vec{b}_1^*\| & 0 & 0 & \dots & 0 \\
\mu_{2,1}\|\vec{b}_1^*\| & \|\vec{b}_2^*\| & 0 & \dots & 0 \\
\mu_{3,1}\|\vec{b}_1^*\| & \mu_{3,2}\|\vec{b}_2^*\| & \|\vec{b}_3^*\| & \ddots & 0 \\
\vdots & \ddots & \ddots & \ddots & \vdots \\
\mu_{d,1}\|\vec{b}_1^*\| & \mu_{d,2}\|\vec{b}_2^*\| & \dots & \mu_{d,d-1}\|\vec{b}_{d-1}^*\| & \|\vec{b}_d^*\|
\end{pmatrix}
$$

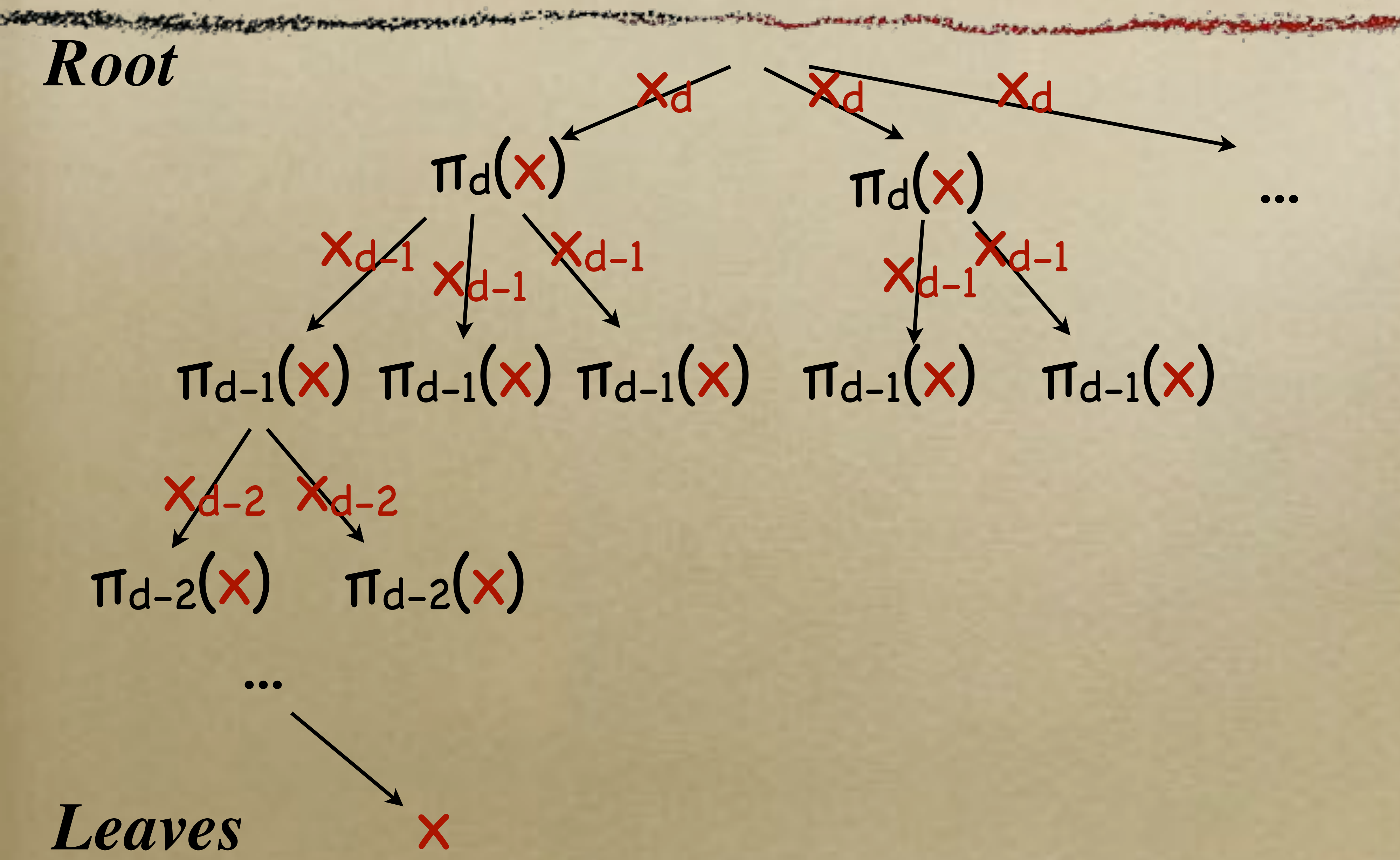
Exhaustive Search

- Let $(b_1,b_2,\ldots b_d)$ be a reduced basis of L.
- Let $x=x_1b_1+x_2b_2+\ldots+x_db_d$ be a shortest vector of L.
- Then $\|\pi_i(x)\|\leq R$ for $1\leq i\leq d$, $R=\|b_1\|$ or $\lambda_1(L)$.
  - $\|\pi_d(x)\|\leq R$ implies: $|x_d|\leq R/\|b_d^*\|$
  - For each value of x_d, $\|\pi_{d-1}(x)\|\leq R$ implies that the integer x_{d-1} belongs to an interval of "small" length.

Enumeration and Triangularization

- Let $x = x_1 b_1 + x_2 b_2 + \ldots + x_d b_d$ be a shortest vector of L.
- Decompose x over the triangular representation of L.
  - Then $\|x\| \leq \|b_1\|$ implies: $|x_d| \leq \|b_1\| / \|b_d^*\|$
  - And so on... each integer x_i belongs to an interval of "small" length.

Enumeration Tree



Enumeration tree

- Depth k contains all projected lattice points $\|\pi_{d+1-k}(y)\|$ ($y \in L$) of norm $\leq$ R.
- The leaves are all $y \in L$ of norm $\leq$ R.
- Enumeration searches the whole tree to compute all leaves, compare their norm to output a shortest vector $x \in L$.

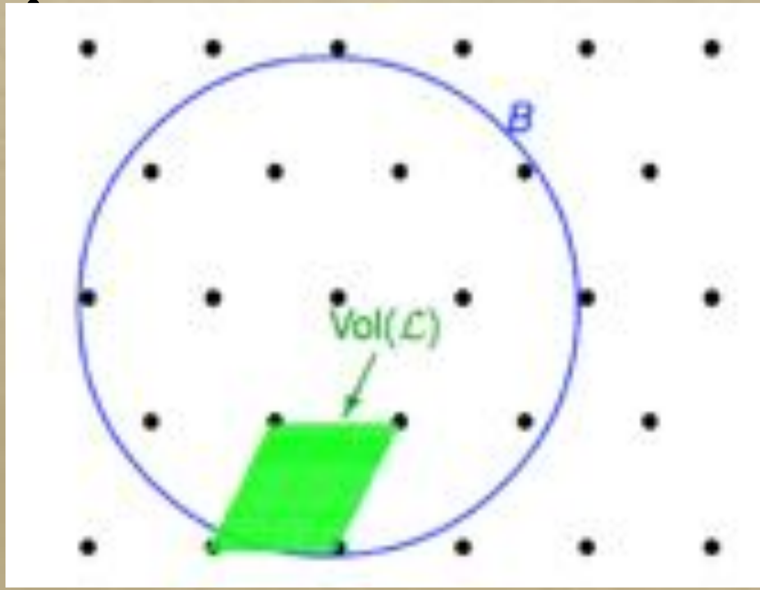
Complexity of Enumeration

- The complexity of enumeration is, up to a polynomial factor, the number of lattice points in all projected lattices inside the centered ball of radius R.
- This number can be upper bounded, but worst-case bounds are typically higher than experimental numbers.

The Gaussian Heuristic

- The volume is the inverse density of lattice points.
- For "nice" full-rank lattices L, and "nice" measurable sets C of $\mathbf{R}^n$:

$$\mathrm{Card}(L \cap C) \approx \frac{\mathrm{vol}(C)}{\mathrm{vol}(L)}$$



Validity of the Gaussian Heuristic

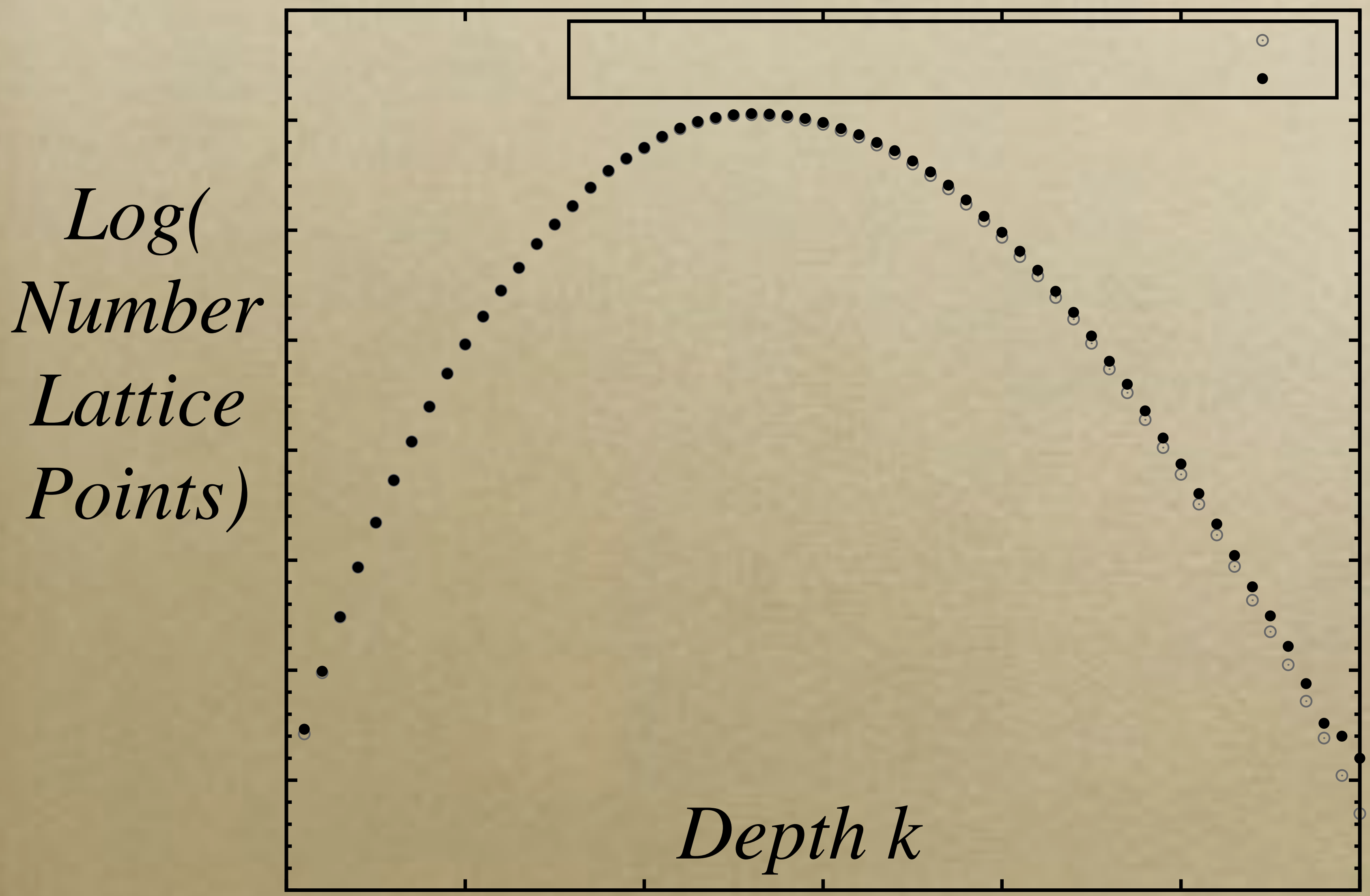
- Easy to prove for arbitrarily large balls: 1/vol(L) = $\lim_{r \longrightarrow \infty}$ (number of lattice points of norm ≤ r)/vol(Ball(0,r))
- If μ(L) is the covering radius,

$$\#(L \cap B(0,R)) \leq \frac{\mathrm{vol}(B(R+\mu(L)))}{\mathrm{vol}(L)}$$

Practical Complexity of Enumeration

- By the Gaussian heuristic, the number of lattice points should be $\approx\sum_{1\leq k\leq d} v_k(R)/\mathrm{vol}(\pi_{d-k+1}(L))$, where $v_k(R)$ is the volume of the k-dim ball of radius R.
- Intuitively, this should be ok, as while as each term is very big.

Accuracy of Gaussian Heuristic



Remark

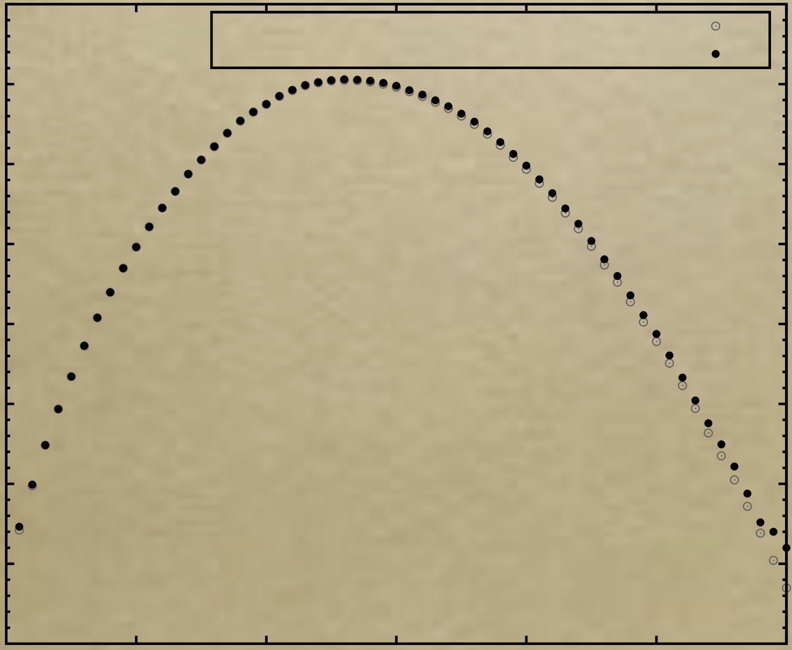
- It is not shocking that the Gaussian heuristic is accurate here: we're estimating the number of "short" vectors in a projected lattice, where the radius is significantly larger than the dim-th root of the volume. This is an exponential number.

Practical Complexity of Enumeration

- By the Gaussian heuristic, the number of lattice points should be $\approx\sum_{1\leq k\leq d} v_k(R)/vol(\pi_{d-k+1}(L))$, where $v_k(R)$ is the volume of the k-dim ball of radius R.
- We can estimate each of this term, using a modelization of reduced bases.

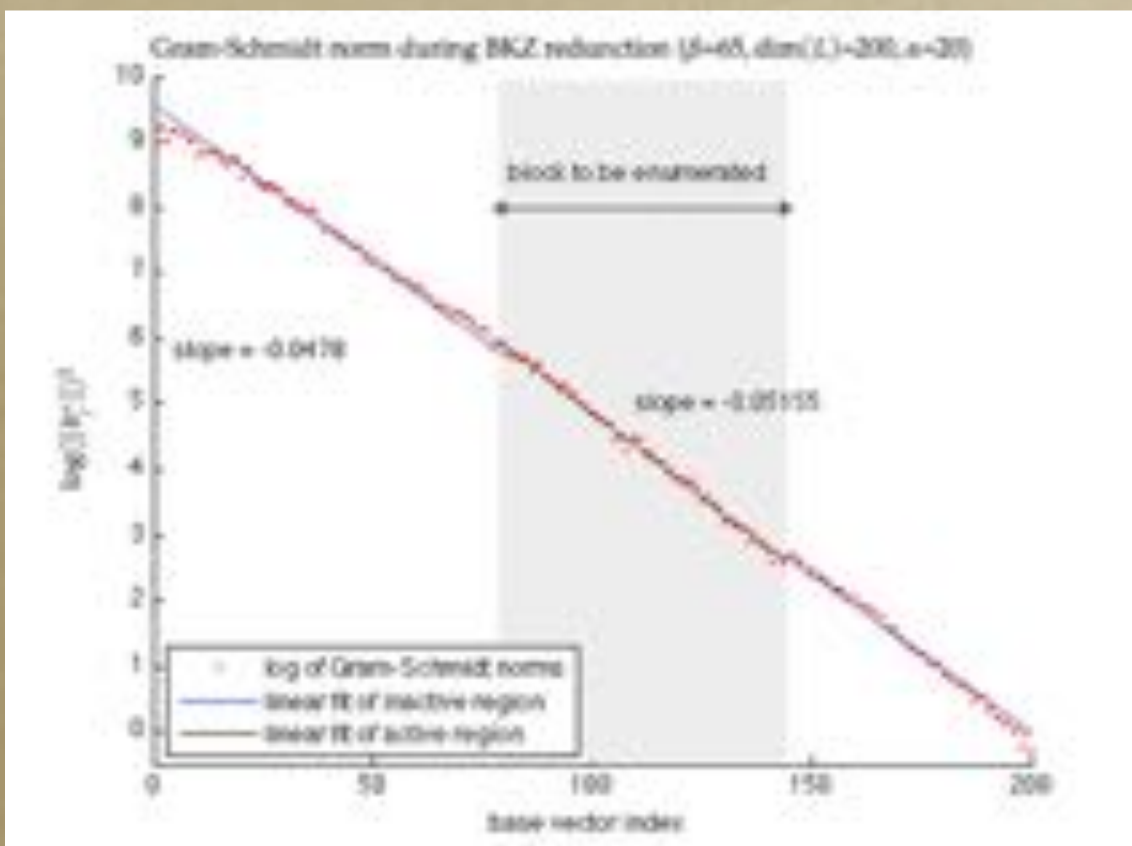
Shape

- For typical reduced bases, the Gram-Schmidt norms decrease geometrically in practice: most of the tree nodes are in middle depths k≈d/2. Their number is super-exponential.



Gram-Schmidt Shape

- Gram-Schmidt log-norms typically form a straight line: this is Schnorr's Geometric Series Assumptions (GSA).



What do we deduce for the Gaussian heuristic?

Take Away

- Enumeration is based on one key idea
  - Projection to decrease the lattice dimension
- Once parameters are fixed, it is possible to reasonably estimate the running time

Optimizing the Basis

- The basis should be chosen to minimize $\sum_{1 \le k \le d} v_k(R)/\mathrm{vol}(\pi_{d-k+1}(L))$ especially for $k \approx d/2$, i.e. to minimize $\mathrm{vol}(b_1,\ldots,b_{d-k}) = \|b_1^*\| \ldots \|b_{d-k}^*\|$.
- In particular, we'd like to minimize $\|b_1^*\| \ldots \|b_{d/2}^*\|$.

Speeding Up Enumeration by Pruning

Speeding Up Enumeration

- Assume that we do not need all $L \cap S$:
  - What if we only need to find one such vector?
  - Can we make enumeration faster?

Enumeration with Pruning

- Input: a lattice L, a ball $S \subseteq \mathbf{R}^n$ and a pruning set $P \subseteq \mathbf{R}^n$.
- Output: All points in $L \cap S \cap P$.

- Started with [ScEu94,ScHo95].

Enumeration with Pruning

- Input: a lattice L, a ball $S \subseteq \mathbf{R}^n$ and a pruning set $P \subseteq \mathbf{R}^n$.
- Output: All points in $L \cap S \cap P$.
- Pros: Enumerating $L \cap S \cap P$ can be much faster than $L \cap S$.
- Cons: Maybe $L \cap S \cap P \subseteq \{0\}$. We get nothing.

Analyzing Pruned Enumeration [GNR10]

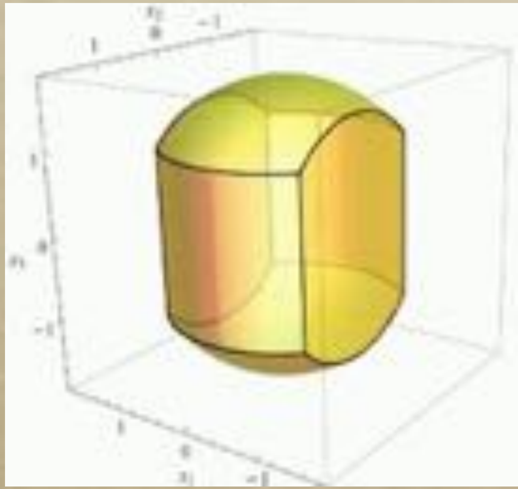
- More sound than previous analyses: enumerating L∩S∩P is deterministic.
- [GNR10] framework:
  - The set P is randomized: it depends on a (random) reduced basis.
  - The success probability is Pr(L∩S∩P $\not\subseteq$ {0}).
  - By the Gaussian heuristic, #(L∩S∩P) « should » be close to vol(S∩P)/covol(L).

Extreme Pruning [GNR10]

- Repeat until success
  - Generate P by reducing a "random" basis.
  - Enumerate($L \cap S \cap P$)
- Even if $\Pr(L \cap S \cap P \not\subseteq \{0\})$ is tiny, what matters is the trade-off: $\mathrm{Cost}(\mathrm{Enum}(L \cap S \cap P)) / \Pr(L \cap S \cap P \not\subseteq \{0\})$

Two Kinds of Pruning

- Continuous Pruning ([GNR10] generalizing [ScEu94,ScHo95]): P is a cylinder intersection.

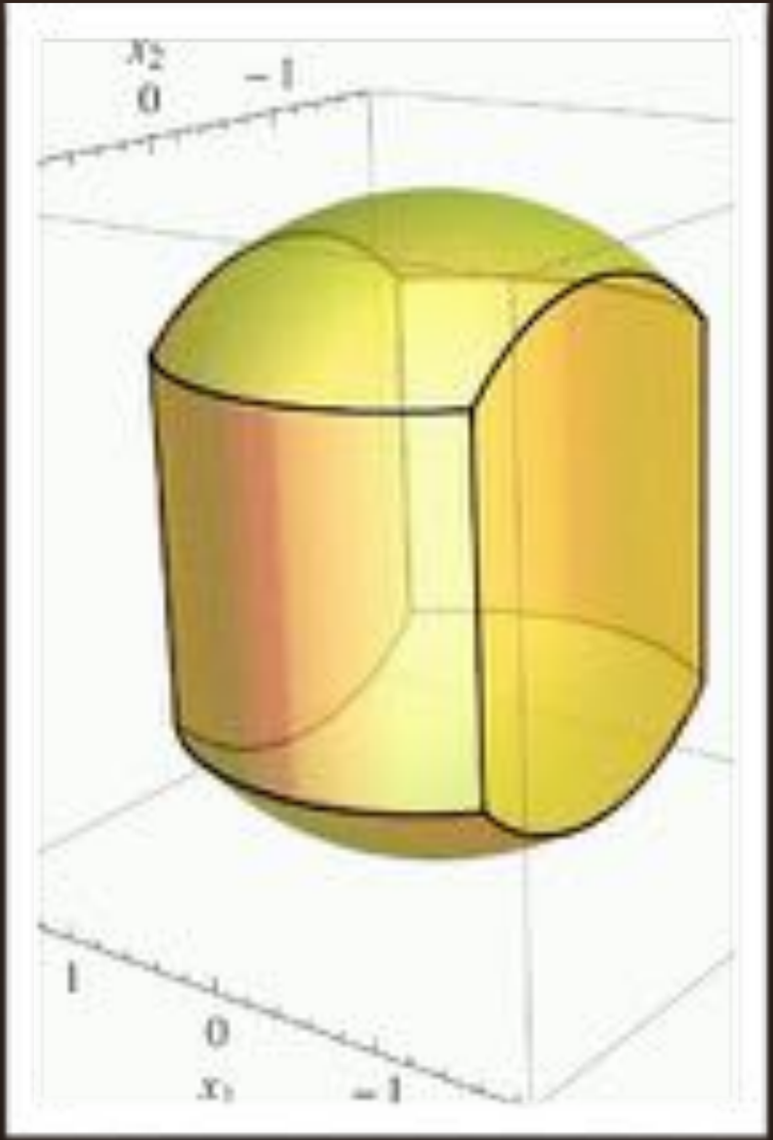


- Discrete Pruning ([AoN17] generalizing [Sc03,FuKa15]): P is a union of cells, in practice a union of boxes.

Take Away

- Pruned enumeration is based on more key idea
  - Slicing the ball in a randomized manner
- Once all parameters are fixed, it is possible to reasonably estimate the running time. But difficult to optimize.

Cylinder Pruning



Cylinder Prutning

- [ScEu94,ScHo95], revisited in [GNR10].
- Idea: random projections are shorter.
- We can prune the gigantic tree.

Pruned enumeration cuts off many branches, by bounding projections.

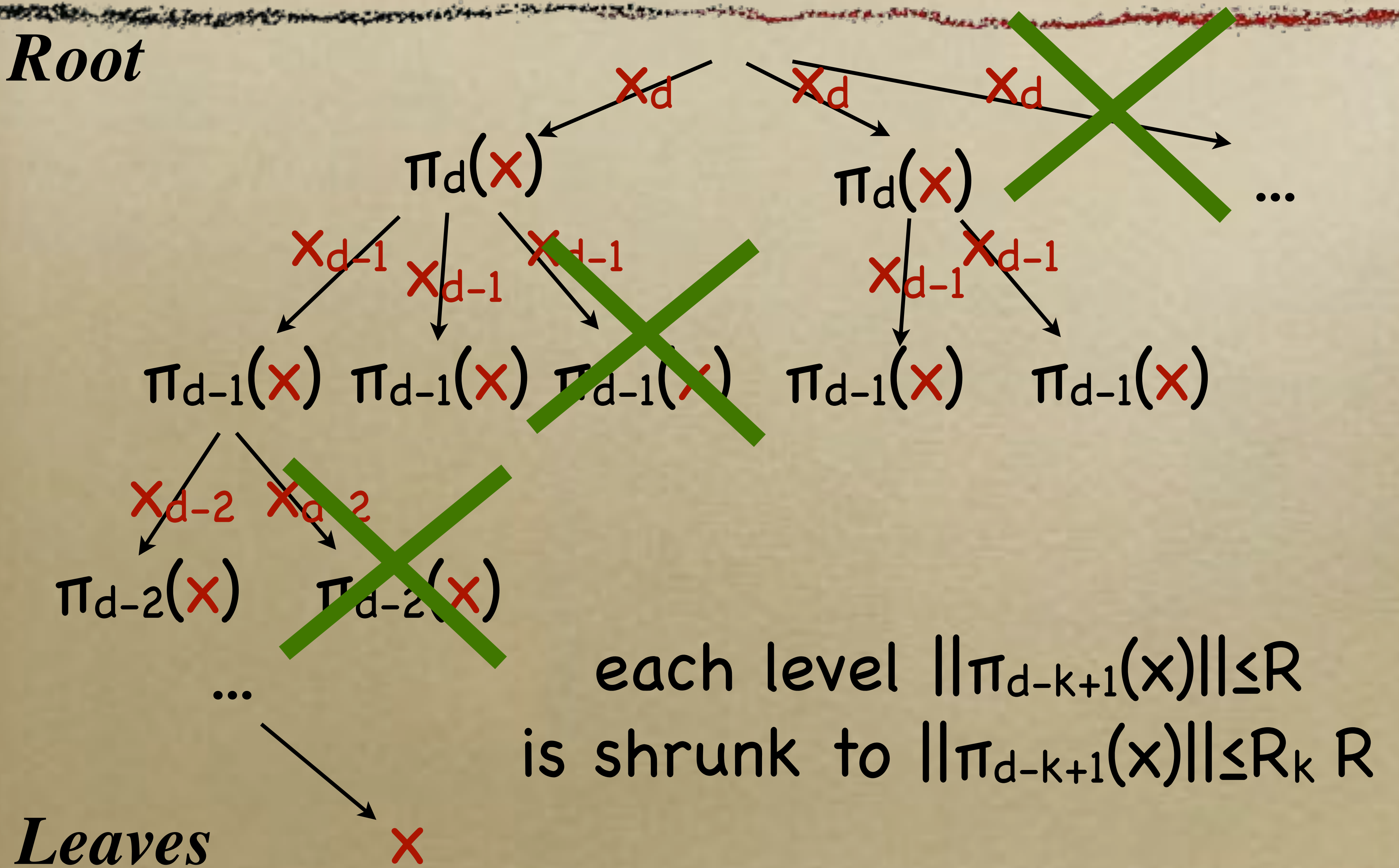
Intuition

- Enumeration says:
  If $||x|| \leq R$, then $||\pi_{d+1-k}(x)|| \leq R$ for all $1 \leq k \leq d$
- But if you choose x at random from the ball of radius R, then its projections $\pi_{d+1-k}(x)$ are likely to be shorter.
- For instance, we would expect $||\pi_{d/2}(x)|| \approx R/\sqrt{2}$.

Cylinder Pruning

- Replace each inequality $||\pi_{d-k+1}(x)|| \leq R$ by $||\pi_{d-k+1}(x)|| \leq R_k R$ for each index k in $\{1,\ldots,d\}$, where $0 < R_k \leq 1$.
- The enumeration tree is pruned with $P = \{x \in \mathbf{R}^d \text{ s.t. } ||\pi_{d-k+1}(x)|| \leq R_k R \text{ for } 1 \leq k \leq d\}$. Again, one searches the tree to find all leaves.
- The algorithm is faster because there are less nodes.

Cylinder-Enumeration Tree



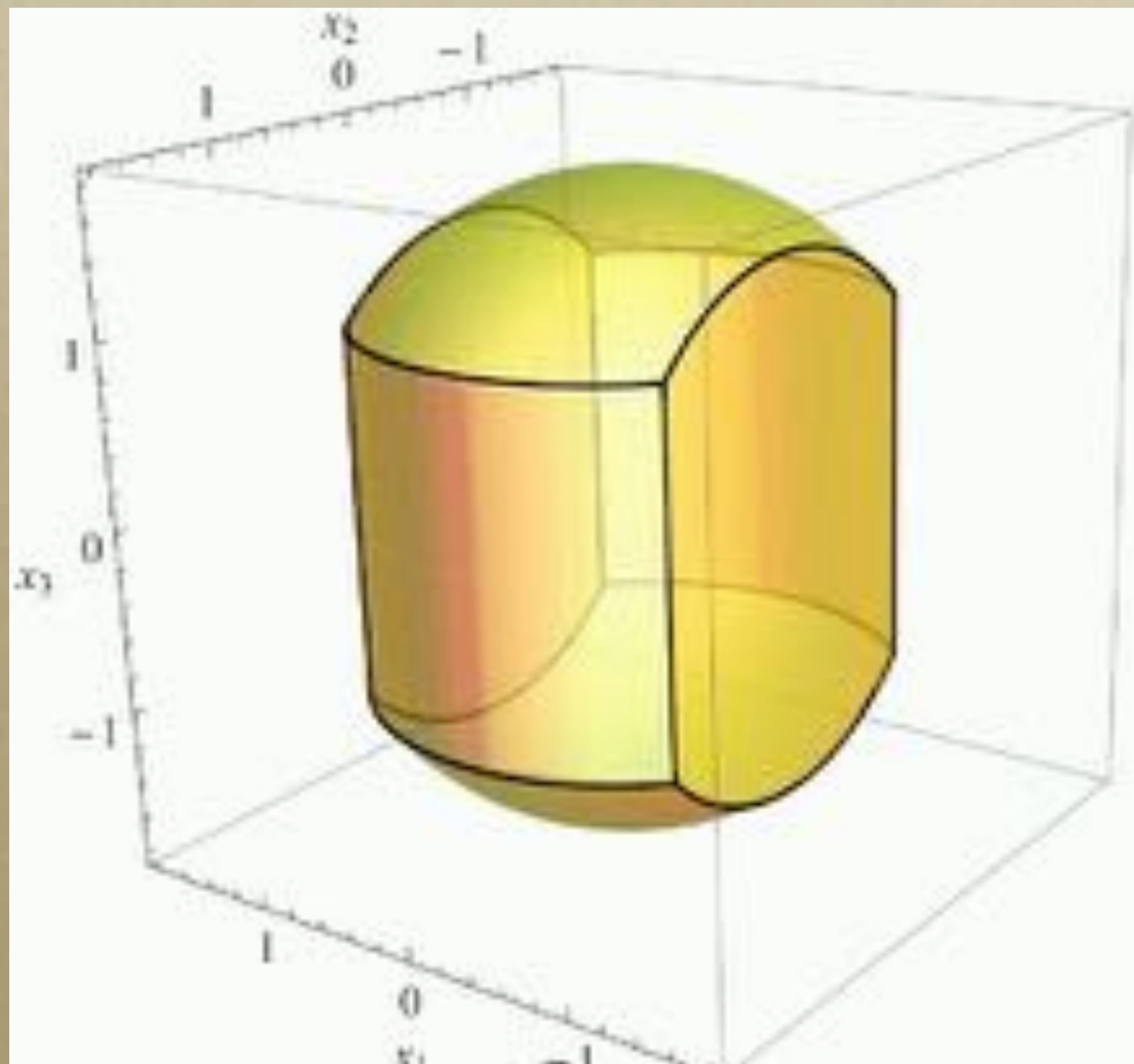
each level $\|\pi_{d-k+1}(x)\| \leq R$

is shrunk to $\|\pi_{d-k+1}(x)\| \leq R_k\, R$

Enumeration with cylinder pruning

- The complexity is, again up to a polynomial factor, a number of lattice points in projected lattices, but instead of balls, we have to consider new sets, whose volume might be harder to compute.

Balls Replaced by Cylinder Intersections



More Precisely

- The k-dimensional ball of radius R, is replaced by: $\{(y_1,\ldots,y_k)\in\mathbf{R}^k$ s.t. for all $1\leq i\leq k$, $y_1^2+\ldots+y_i^2 \leq R_i^2 \times R^2\}$.
- Its volume is $V_k(R)$ times the probability P_k that for $(y_1,\ldots,y_k)$ chosen uniformly at random from the unit ball, $y_1^2+\ldots+y_i^2 \leq R_i^2$ for all $1\leq i\leq k$.

In other words

- The heuristic complexity of enumeration $\Sigma_{1\leq k\leq d}\, v_k(R)/vol(\pi_{d-k+1}(L))$ is reduced to $\Sigma_{1\leq k\leq d}\, v_k(R)P_k/vol(\pi_{d-k+1}(L))$.
- At depth k, the number of nodes is reduced by the multiplicative factor P_k.

Remark

- For fixed i, the probability that for $(y_1,\ldots,y_k)$ chosen uniformly at random from the unit ball, $y_1^2+\ldots+y_i^2 \le R_i^2$ is easy to compute.
- But the joint probability P_k seems hard in general.

Technical Problem [GNR10]

- To analyze and select good parameters for continuous pruning, we need to estimate the volume of:
  - $\{(y_1,\ldots,y_n) \in \mathbf{R}^n$ s.t. for all $1 \leq k \leq n$, $y_1^2+\ldots+y_k^2 \leq R_k^2\}$ for given $R_1, R_2, \ldots, R_n$.
  - This can be done efficiently thanks to the Dirichlet distribution and well-chosen polytopes.

Special case: Linear Pruning

- An interesting easy case: $R_i=\sqrt{(i/d)}$.
- Then we can prove:
  - $(k/d)^{k/2} \leq P_k \leq k(k/d)^{k/2}$
  - Thus, for $k \approx d/2$, $P_k \approx 1/2^{d/4}$

Special cases: The Even Case

- k even and $R_1=R_2$, $R_3=R_4$,...,$R_{k-1}=R_k$.
- If $(y_1,\ldots,y_k)$ is chosen uniformly at random from the unit ball, then $(y_1^2+y_2^2,\ y_3^2+y_4^2,\ldots,\ y_{k-1}^2+y_k^2)$ has uniform distribution over a simplex, due to the Dirichlet distribution.
- Then computing P_k is reduced to computing easy integrals:

$$\int_{y_1=0}^{t_1}\int_{y_2=y_1}^{t_2}\ldots\int_{y_\ell=y_{\ell-1}}^{t_\ell} dy_\ell \ldots dy_1$$

Special cases: The Odd Case

- k odd and $R_1=R_2$, $R_3=R_4,\ldots,R_{k-2}=R_{k-1}, R_k$.
- Then computing P_k is reduced to computing (slightly more complex) easy integrals:

$$\int_{y_1=0}^{t_1}\int_{y_2=y_1}^{t_2}\ldots\int_{y_\ell=y_{\ell-1}}^{t_\ell}\sqrt{1-y_\ell}\,dy_\ell\ldots dy_1$$

General Case

- The probability P_k can be computed numerically by Monte Carlo sampling:
  - Pick many $(y_1,\ldots,y_k)$ at random from the unit ball.
  - Count how many times $y_1^2+\ldots+y_i^2 \leq R_i^2$ for all $1 \leq i \leq k$.
- This is inefficient if P_k is very small. To improve efficiency, one can replace balls by smaller sets of known volume.

General Case

- The odd and even cases allow to compute efficiently an upper bound and a lower bound for any bounding function.
- Using similar integrals, one can in fact also compute an arbitrarily good approximation using efficient Monte-Carlo sampling.

Optimizing the Basis

- The basis should be chosen to minimize $\Sigma_{1\leq k\leq d}\, v_k(R)P_k/\mathrm{vol}(\pi_{d-k+1}(L))$ especially for $k\approx d/2$, i.e. to minimize $\mathrm{vol}(b_1,\ldots,b_{d-k}) = \|b_1^*\|\ldots\|b_{d-k}^*\|$ because P_k does not depend on P.
- In particular, we'd like againto minimize $\|b_1^*\|\ldots\|b_{d/2}^*\|$.

Discrete Pruning

Lattice Partitions

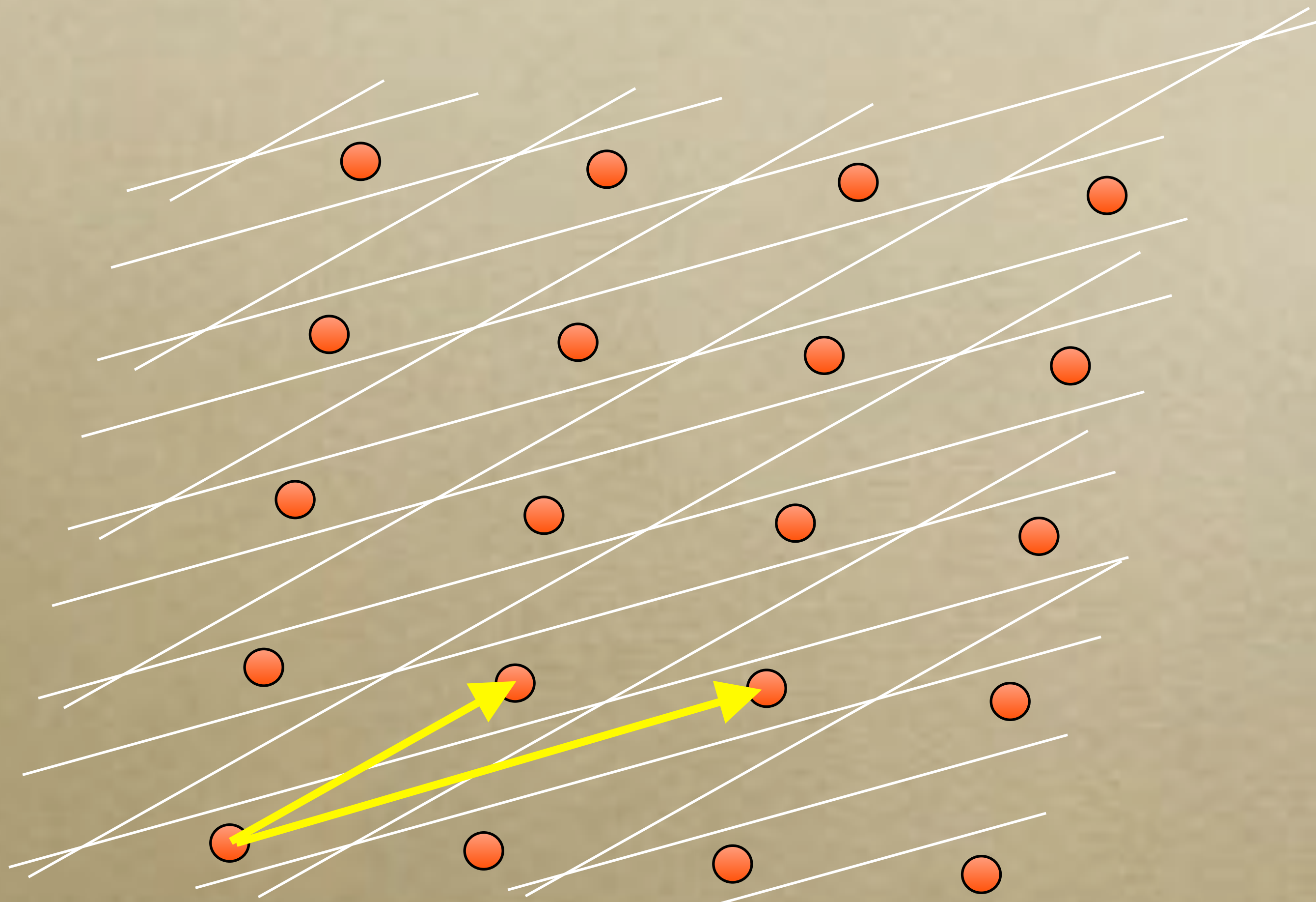
- Any partition of $\mathbf{R}^n = \cup_{t \in T} C(t)$ into countably many cells (T is countable) s.t.:
  - the cells are disjoint: $C(i) \cap C(j) = \varnothing$
  - each cell contains one and only one lattice point which can be found efficiently: given $t \in T$, one can efficiently compute $L \cap C(t)$.

Lattice Enumeration with Discrete Pruning [AoN17]

- Repeat until success
  - Select $P=\cup_{t\in U} C(t)$ for some finite subset $U \subseteq T$.
  - Enumerate($L \cap S \cap P$) by enumerating all $C(t) \cap L$ where $t \in U$.
- The running time is essentially $\#U$ / $\Pr(L \cap S \cap P \not\subseteq \{0\})$: we just need to calculate $vol(S \cap C(t))$.

Fundamental Domain from Bases

Fundamental Domain from Bases

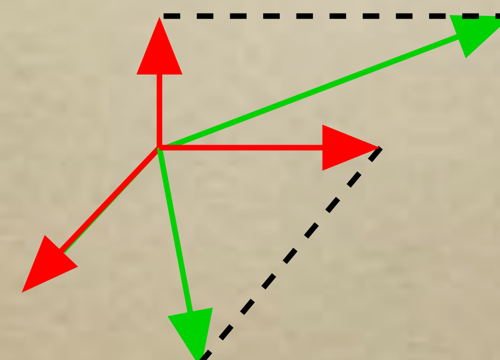


Ex: Fundamental Domains

- A fundamental domain of a lattice L is a measurable subset $D \subseteq \mathbf{R}^n$ s.t. $\mathbf{R}^n = \cup_{v \in L} (v+D)$ and the interiors of v+D are disjoint.
- Then we can select $T = \mathbf{Z}^n$ and $C(t) = tB+D$ where B is a lattice basis, except that the C(t)'s may overlap at the frontier. However, we already know the lattice point tB.

Gram-Schmidt

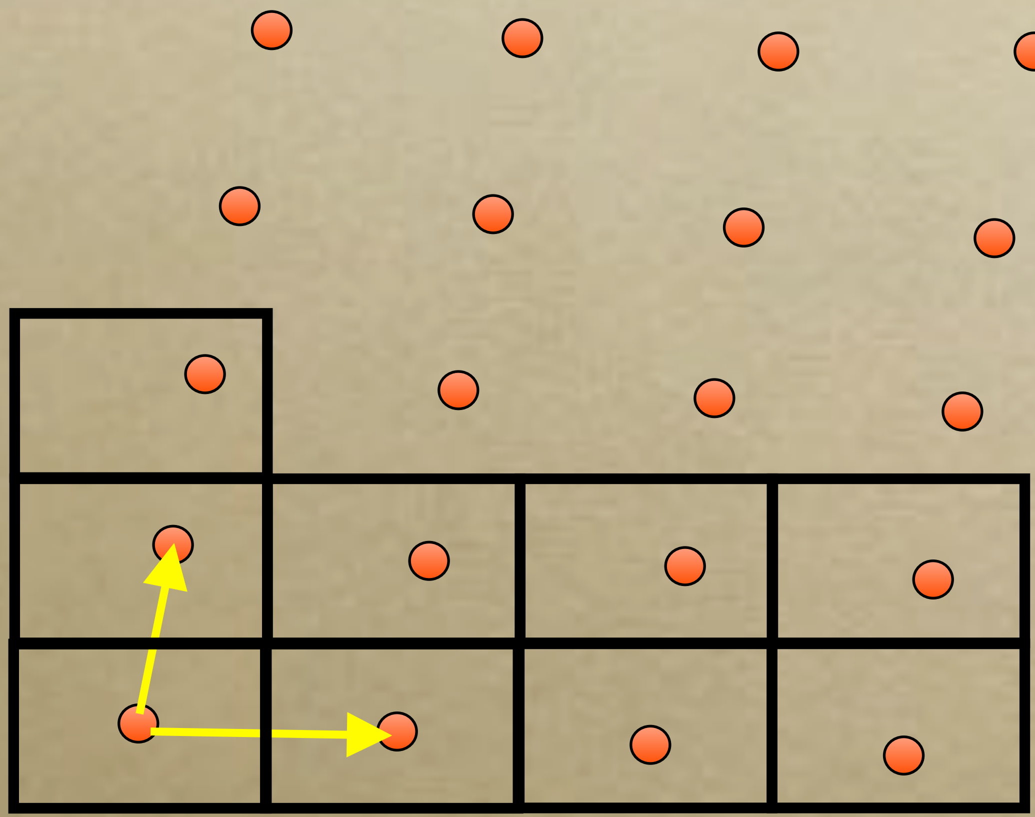
- Let $b_1,\ldots,b_n \in \mathbf{R}^m$.
- Its Gram-Schmidt Orthogonalization is $b_1^*,\ldots,b_n^* \in \mathbf{R}^m$ defined as:
  - $b_1^* = b_1$
  - For $2 \leq i \leq n$, b_i^* = component of $b_i b_i$ orthogonal to $b_1,\ldots,b_{i-1}$ = projection of b_i over $\mathrm{span}(b_1,\ldots,b_{i-1})^\perp$



Ex: Fundamental Domains

- To avoid this problem, we choose a set which is a fundamental domain for two lattices!
  - Let $(b_1,\ldots,b_n)$ be a basis of L and $(b^*_1,\ldots,b^*_n)$ be its Gram-Schmidt vectors.
  - Then $D=\{\sum_i x_i b^*_i$ s.t. $-1/2 \le x_i \le 1/2\}$ is a fundamental domain for both L and the Gram-Schmidt lattice $L(b^*_1,\ldots,b^*_n)$.
- Then we can select $T=\mathbf{Z}^n$ and $C(t) = tB^*+D$.

The Gram-Schmidt Fundamental Domain

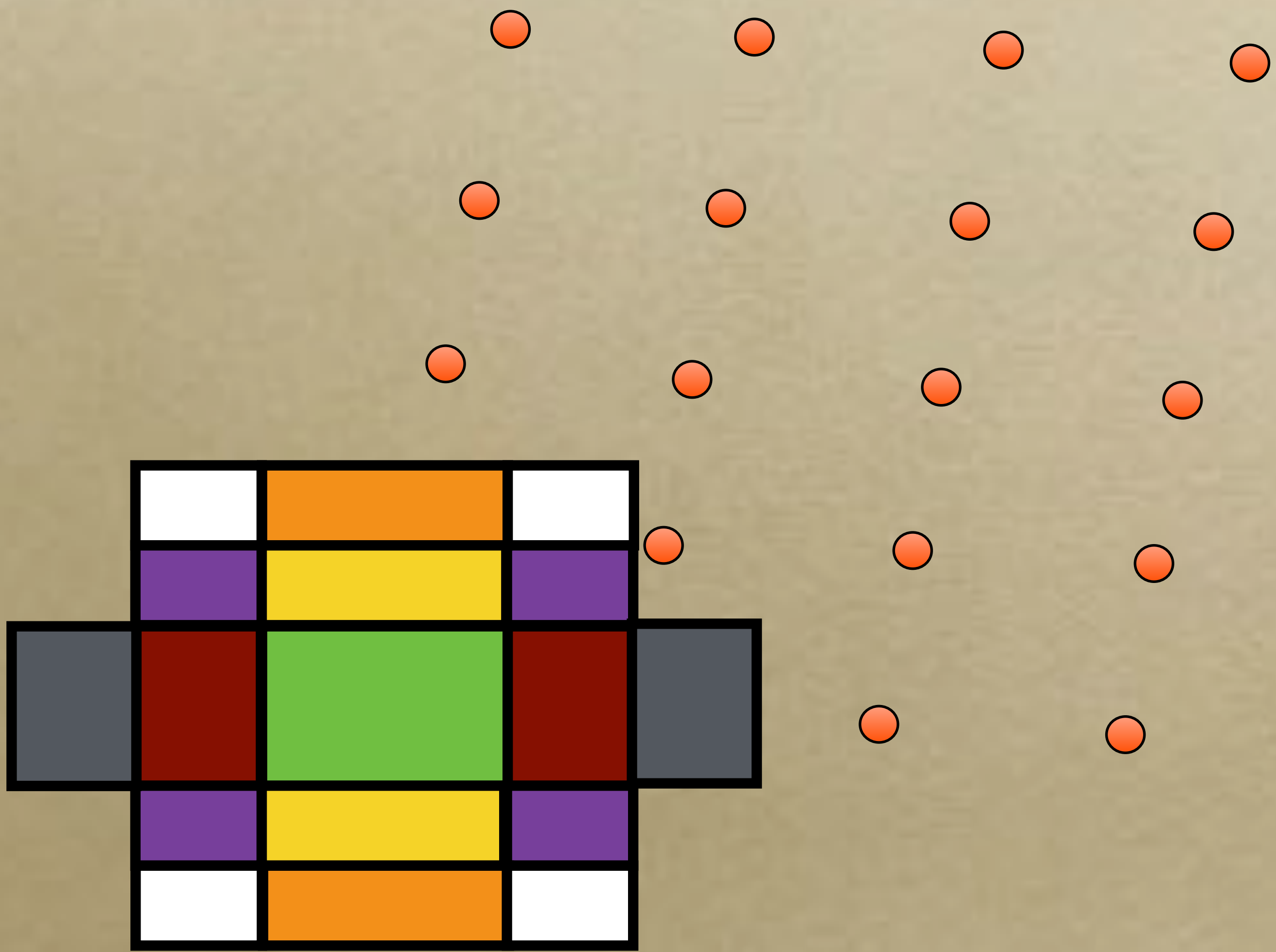


Ex: Partition with Natural Integers

- [FuKa15] implicitly used a variant of this partition: $T=\mathbf{N}^n$ and $C((t_1,\ldots,t_n))$ is the parallelepiped $\{\sum_i x_i b^*_i$ s.t. $-(t_j+1)/2<x_j\leq -t_j/2$ or $t_j/2<x_j\leq(t_j+1)/2\}$ whose volume is covol(L). Here, the b^*_i's are the Gram-Schmidt vectors of a lattice basis.

The Gram-Schmidt Partition

The « Natural » Partition

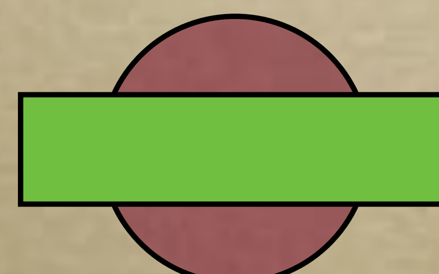


Discrete Pruning

- Both [Sc03] and [FuKa15] use the natural partition with some finite set J:
  - [Sc03] uses essentially $J=0^{n-k-1}\{0,1\}^k 1$ so $\#J=2^k$.
  - [FuKa15] uses a J constructed by an algorithm and experiments: $\#J=5\times10^7$.
- Instead, we suggest to use the J with the maximal $\mathrm{vol}(S \cap C(t))$.

Is it Over?

- This discrete pruning is very easy to implement.
- But there is one technical issue: to estimate the success probability, we need to approximate vol(S∩C(t)) for many t's where:
  - S is a ball
  - C(t) is a box, or a union of symmetric boxes.



Intersection of a Ball with a Box

- Let B=unit-ball and $H=\Pi_i [\alpha_i, \beta_i]$ be a box.

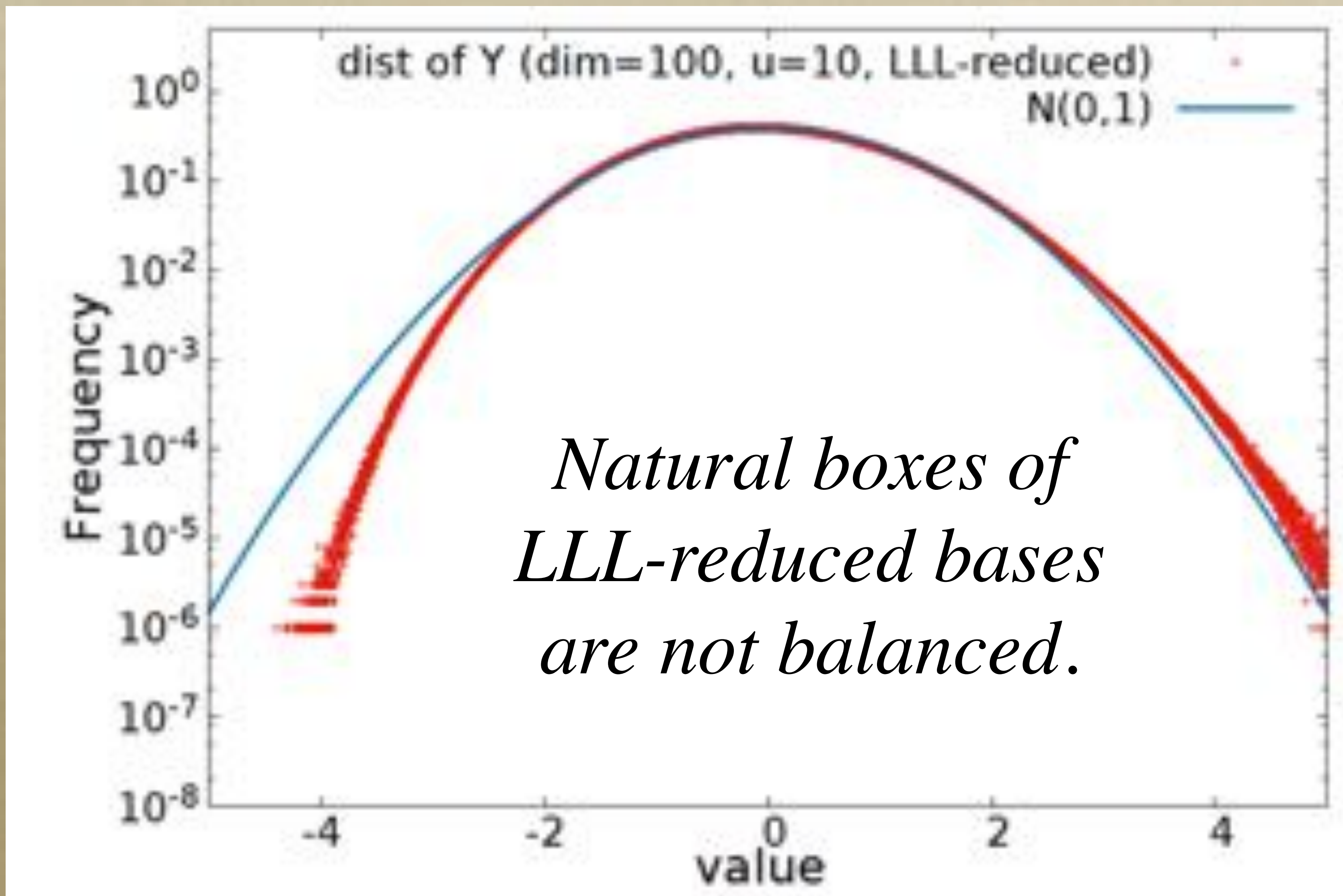
  Compute $vol(S \cap H)$.

- Asymptotic formula from the central limit theorem:
  - Th: If H is 'balanced', $(||x||^2 - E_{y \in H}(||y||^2)) / \sqrt{V_{y \in H}(||y||^2)}$ converges to $N(0,1)$ when x is uniform over H.

CLT vs Natural Boxes

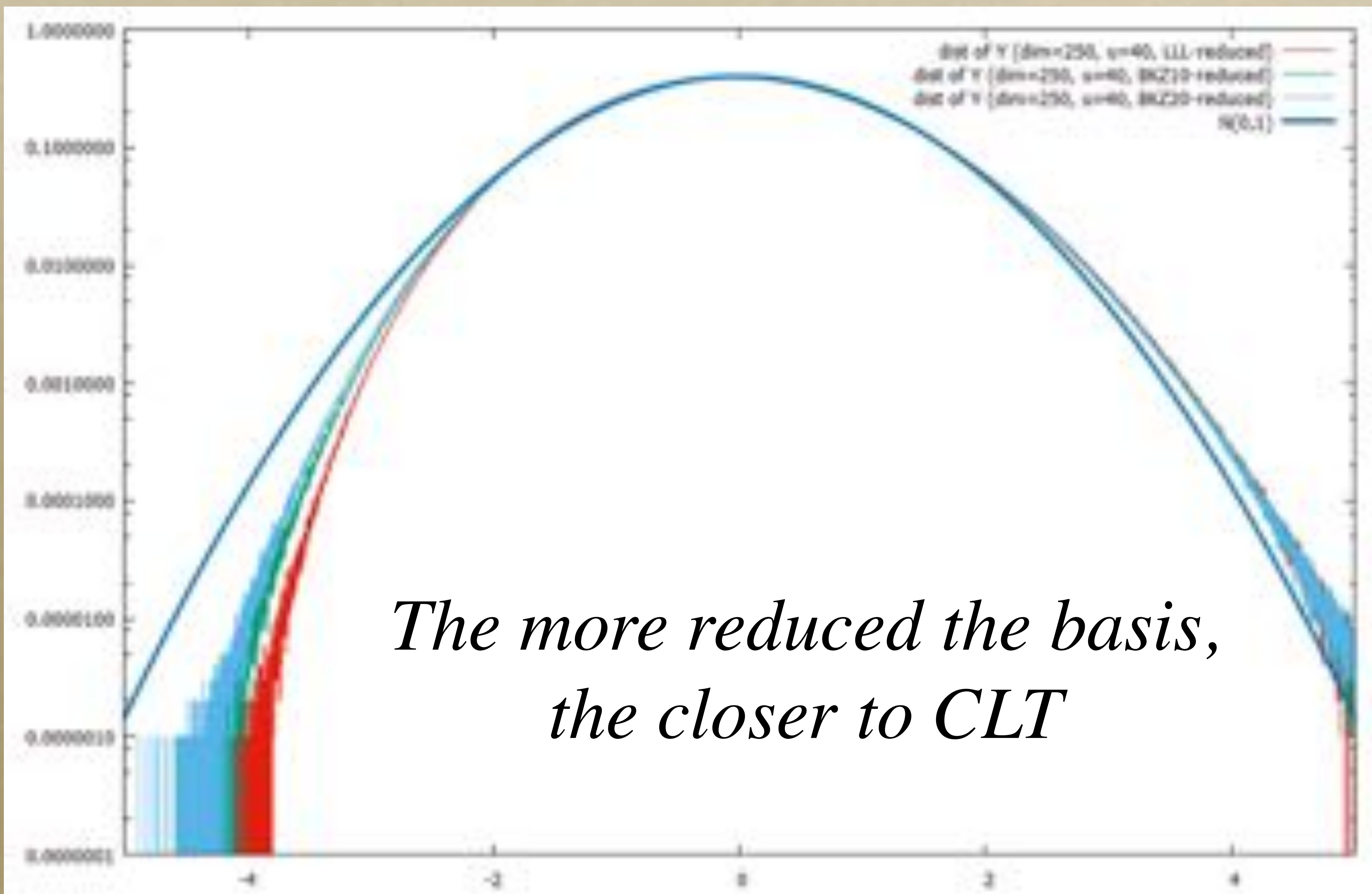
- Let B=unit-ball and $H=\Pi_i [\alpha_i, \beta_i]$ be a box.
- In our case, the natural box H is not balanced, because the b_i^* typically decrease geometrically, but the more reduced the basis, the closer to CLT.

CLT vs Natural Boxes



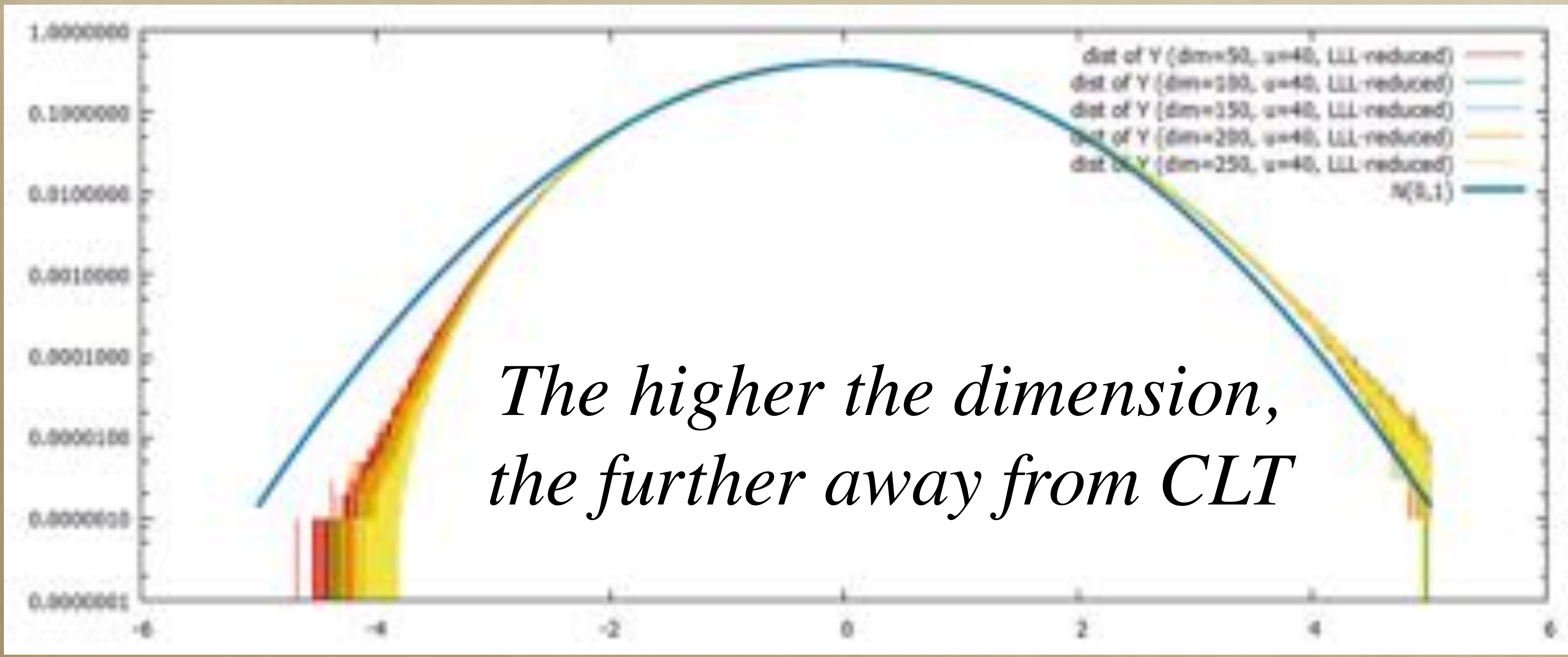
Natural boxes of LLL-reduced bases are not balanced.

CLT vs Natural Boxes

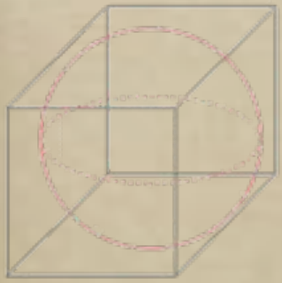


The more reduced the basis,
the closer to CLT

CLT vs Natural Boxes



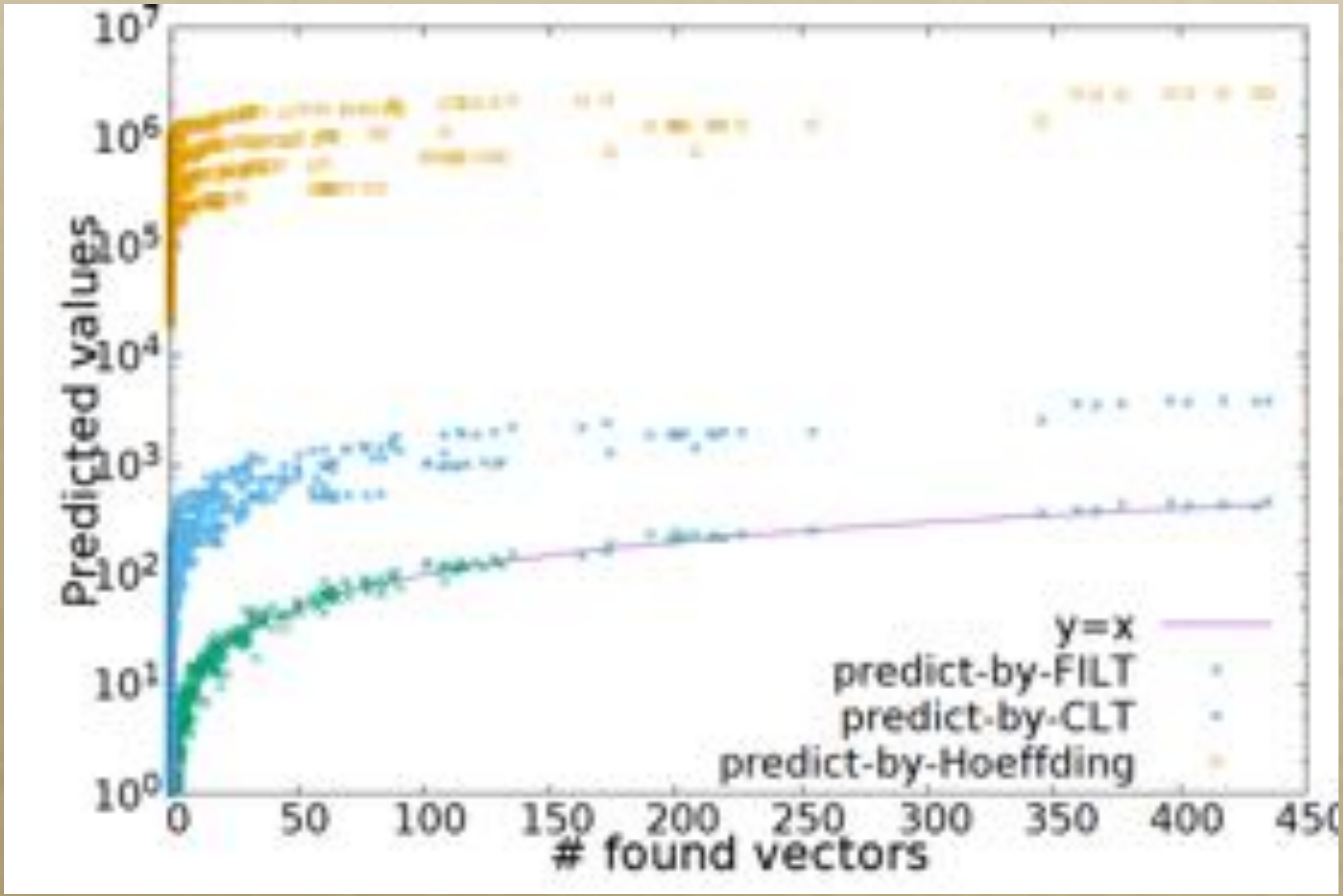
The higher the dimension,
the further away from CLT



Intersection of a Ball with a Box

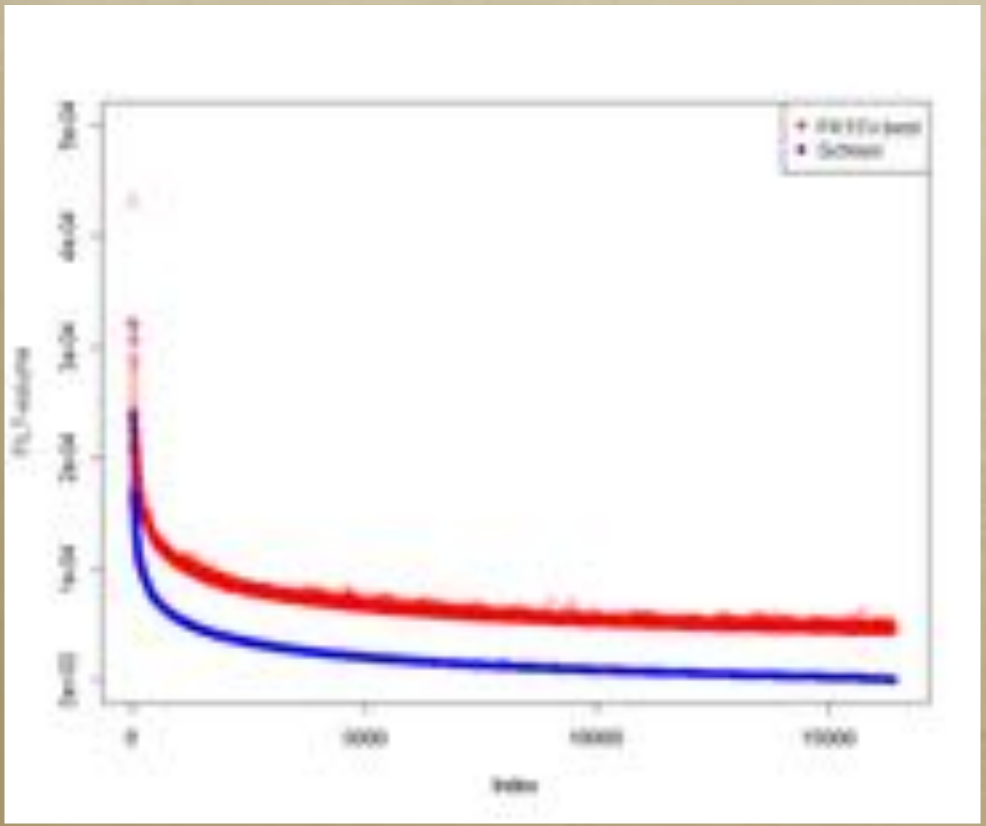
- Let B=unit-ball and $H=\Pi_i [a_i,b_i]$ be a box. Compute vol($S \cap H$).
- We obtain two exact formulas as infinite series, by generalizing [CoTi1997] based on Fourier transforms and Fourier series.
- But in practice, our fastest method uses [Hosono81]'s Fast Inverse Laplace Transform: less than 1s in dim 100.

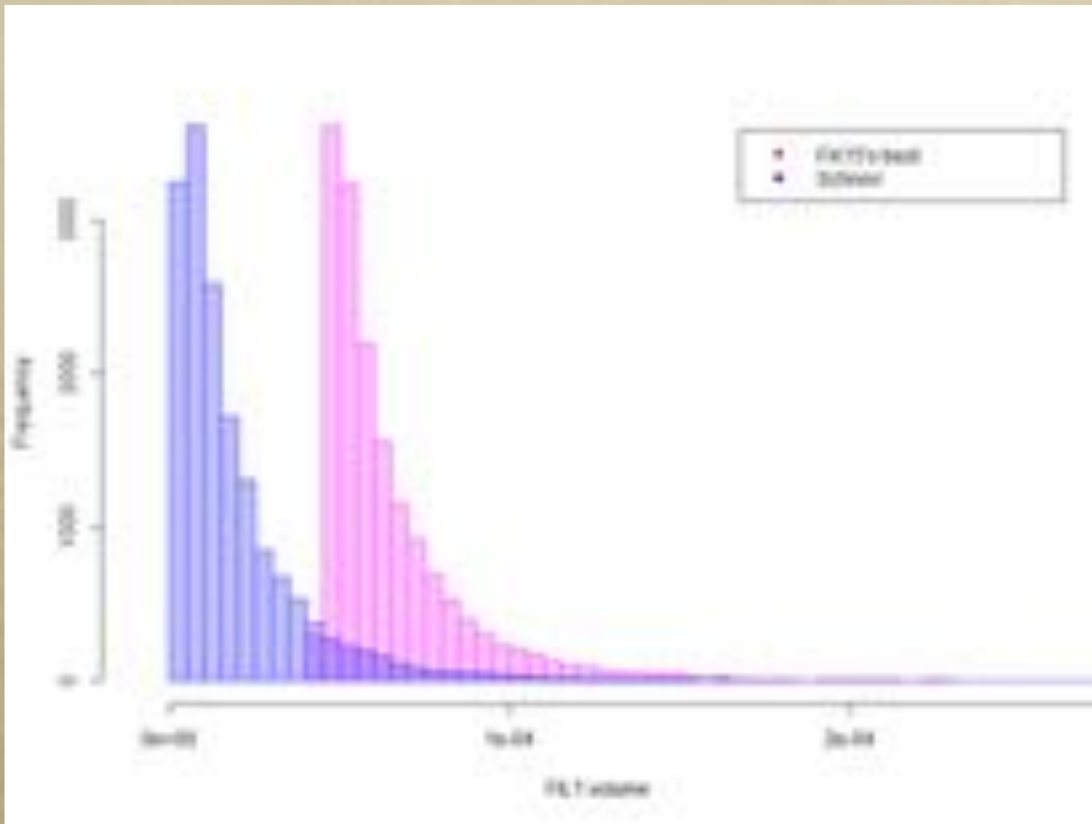
Accuracy of Predictions



Very good predictions

[Schnorr03] vs [FuKa15]



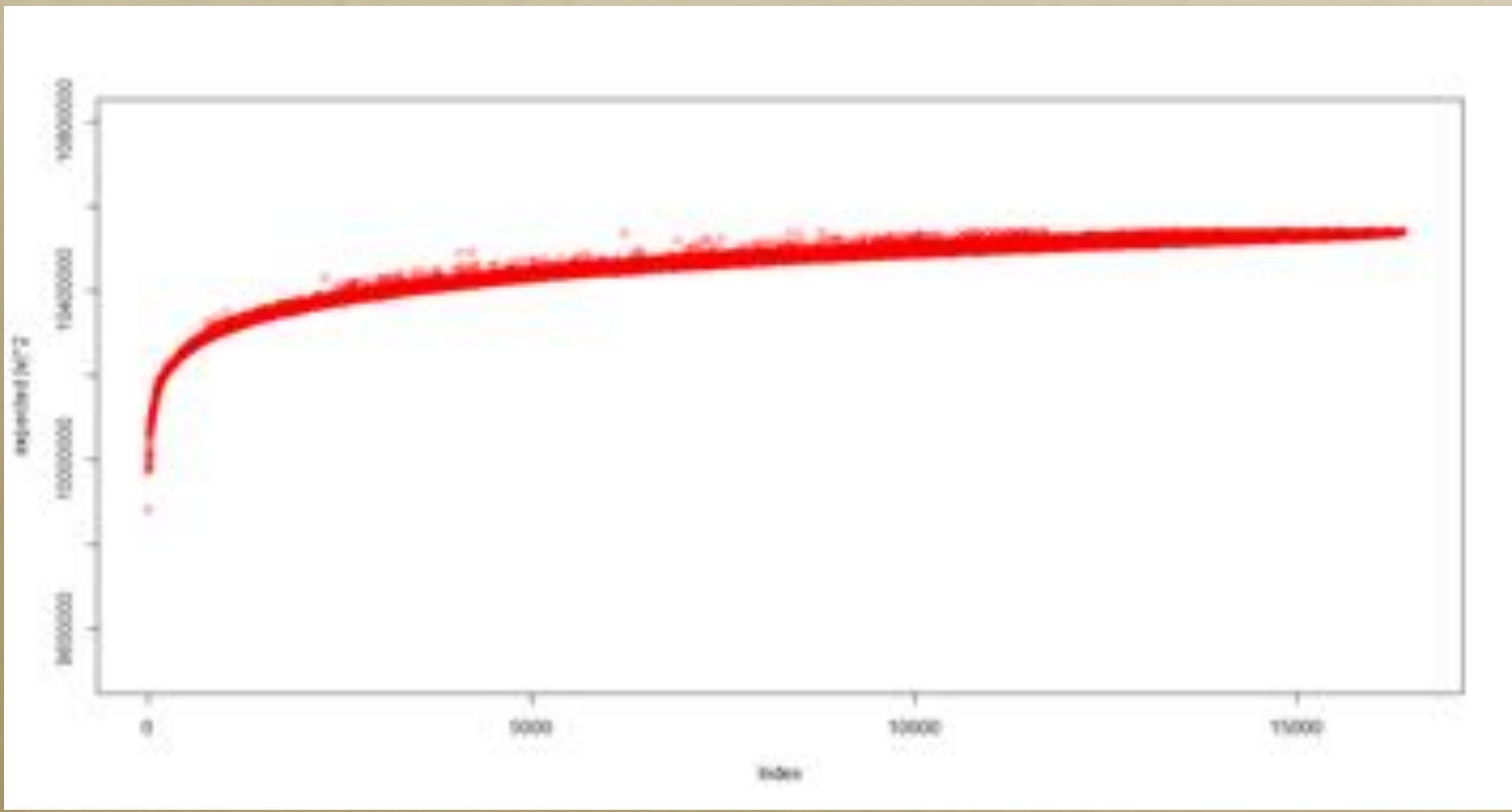


Distribution of vol(S∩C(i))

Heuristics For Selecting Cells

- The exact computation of vol(S∩H) is « slow ». But there is a good heuristic method to select good cells: if $H=C((t_1,\ldots,t_n))$, $E_{x\in H}(||x||^2) = \sum_j (3t_j^2+3t_j+1)||b_j^*||^2/12$.
- Finding all $(t_1,\ldots,t_n)$ minimizing $E_{x\in H}(||x||^2)$ is finding the closest lattice points in the GS lattice inside the positive quadrant. This is very fast because that lattice has an orthogonal basis.

Correlation Between Expectation and Volume



The largest-volume cells

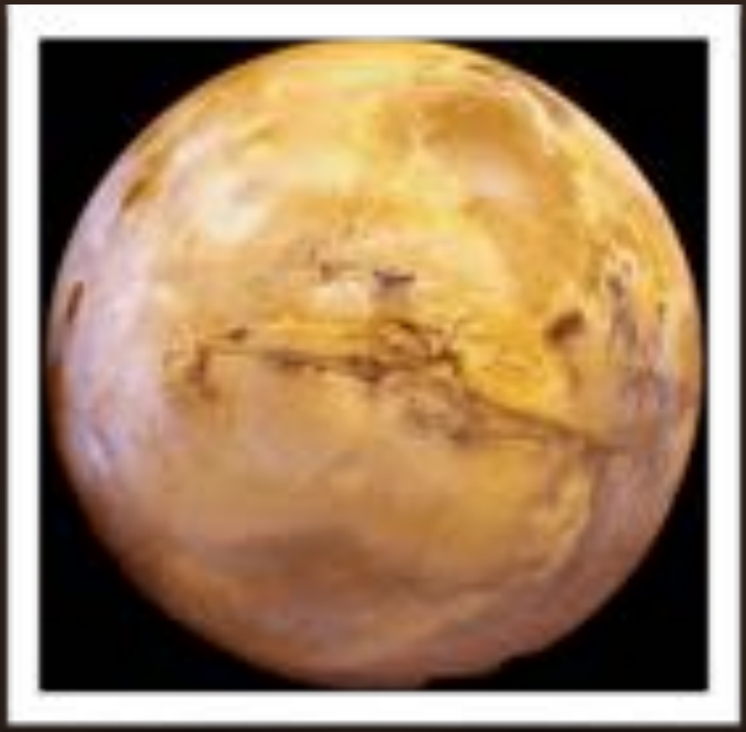
Sums of Volumes by Statistical Inference

- We can compute vol(S∩C(t)), but we would like to do it for millions of t's to approximate $\sum_{t \in U} \mathrm{vol}(S \cap C(t))$.
- So we ``select" say a few thousands cells and… extrapolate!
  - We can get very small errors in practice, say ≤ 1%.

Optimizing the Basis

- The basis should be chosen to minimize vol(S∩C(t)) for our tags t. Heuristically, this may be the same as minimizing $E_{x\in H}(||x||^2) = \sum_j (3t_j^2+3t_j+1)||b_j^*||^2/12$.
- Thus, we may want to minimize $\sum_j ||b_j^*||^2$.
- The best bases for discrete pruning may not be the best bases for cylinder pruning.

Conclusion



Conclusion

- Enumeration is the most effective lattice algorithm in practice to find extremely short vectors. It can also be used to approximate with small factors.
- But it requires pruning, whose main technical tool is the ability to approximate volumes of certain bodies: cylinder intersections or box-ball intersections.

Open Problems

- Asymptotically, what is the best form of pruning?
- Are there other efficient forms of pruning, other than cylinder pruning and discrete pruning?
- Cylinder pruning and discrete pruning can be mixed: is it more efficient?

Conclusion

- We introduced enumeration with discrete pruning, which is an alternative generalized geometric description of random sampling [Sc03,BuLu06,FuKa15].
- It can be analyzed in the same way as [GNR10] for enumeration with continuous pruning: better assumptions, accurate predictions and hopefully, better parameters.

Thank you for your attention...

Any question(s)?