Randomizes Incremental Construction and Random Sampling

- Random Incremental Construction
  - Introduction
  - An example
- Random Sampling
  - Introduction
  - An example
- Linear Programming
  - Basic Facts
  - Random Incremental Construction Algorithm
  - Random Sampling Algorithm

Random Incremental Construction

Input: A set of n objects.

Output: Something S depending on the objects.

Algorithm:

- Randomly permute the input set.
  Let $p_1, \ldots p_n$ be the elements indexed in the random order
- Iteratively compute S_i, the "something" for the set $\{p_1, \ldots p_i\}$

The Algorithm

Input: Two sets of points P and S with $|S| \leq 3$.

Invariant: The $SED(P \cup S)$ has S in its boundary.

Output: $SED(P, S)$: SED for P with S on its boundary, provided such a disk exists.

Algorithm: $SED(P, S)$:

1. if $|S| = 3$ return ball defined by S.
2. if $|P| = 2$ return $B((p + q)/2, \|pq\|/2)$, where $P = \{p, q\}$.
3. chose $p \in P$ at random
4. $B := SED(P \setminus \{p\}, S)$.
5. if $p \in B$ return B.
6. return $SED(P \setminus \{p\}, S \cup \{p\})$.

Linear Programming Facts

$$\begin{array}{lrcl} \min & c^T x & & \\ s.t. & Ax & \leq & b \end{array}$$

- We denote the dimension of the space by d.
- m denotes the number of inequalities in $Ax \leq b$.
- We always assume
  - boundedness of the polyhedron
  - Each vertex is defined by exactly d constraints.
  - The minimum is a unique vertex of the polytope.
- There is an algorithm (Simplex) that solves a LP with at most $6d^2$ constraints in $O(d^{d/2+a})$.

Incremental Linear Programming

Input: A set of constraints H and a set of fixed constraints F.

Output: The optimum vertex

Algorithm: SeidelLP(H, F)

- if $|F| = d$, output Simplex(F).
- if $|H \cup F| \leq 6d^2$, output Simplex($H \cup F$)
- Pick a random constraint $h \in H \setminus F$.
- Recursively find $x =$ SeidelLP$(H \setminus \{h\}, F)$.
- if x does not violate h return x.
- Recursively find $x =$ SeidelLP$(H \setminus \{h\}, F \cup \{h\})$

Running Time

- In the following d stands for the number of constraints we need to find to define the optimal vertex.
- The probability the we make two recursive calls is d/m, since d of m constraints define the optimal vertex.
- Number of Simplex calls:
  $B(m, d) = B(m-1, d) + \frac{d}{m} B(m-1, d-1)$
- Solution $B(m, d) \leq md!$.
- Running time without calls of Simplex:
  $T(m, d) = T(m-1, d) + O(d) + \frac{d}{m} T(m-1, d-1)$.
- Solution $T(m, d) \leq cmd!$, for some constant c.

Correctness

Fact 1:
If $LP(H \setminus h) \neq LP(H)$ then h is binding at the optimal vertex.

- Since there is always an optimal point which is a vertex of the polyhedron.

Thus we know that h is one of d constraints that define the optimal vertex.

Discussion

- In fixed dimension, the algorithm has running time $O(m)$.
- The dependency on d (i.e. $d!$ calls of Simplex) is extremely bad.

An Algorithm based on Random Sampling

The algorithm consists of two parts:

Clarkson 1:

- Has expected $O(d^2 m)$ arithmetic operations
- Expected $2d$ linear programs with $3d\sqrt{m}$ constraints are solved with Clarkson 2.

Clarkson 2:

- Requires $O(d^2 m \log m)$ operations.
- Expected $6d \ln m$ calls of Simplex with at most $6d^2$ constraints.

Clarkson 1

Input: A set H of linear constraints, objective function c.

Output: Optimal point x^*.

Algorithm: Clarkson1(H)

- Let $r = d\sqrt{m}, G = \emptyset$
- Repeat:
  - Choose random $R \in \binom{H}{r}$
  - Compute $x^* =$ Clarkson2$(G \cup R)$
  - Let $V = \{h \in H \mid x^* \text{ violates } h\}$
  - If $|V| \leq 2\sqrt{m}$ then $G = G \cup V$ (Augmentation)
- Until $V = \emptyset$
- Return x^*

Overall Running Time

- The algorithm Simplex is called expected $12d^2 \ln m$ times
- The expected running time is
  $O(d^2 m + 2d(d^2 3d\sqrt{m} \log(3d\sqrt{m})) + d \ln m'\text{Simplex}) =$
  $O(d^2 m + d^4\sqrt{m} \log m + d \ln m'\text{Simplex}) =$
  $O(d^2 m + d^2 \ln m\text{Simplex})$.
  Note that the middle term of the second line is dominated either by the first term or by the last term for all choises of d and m.

The Sampling Lemma

Let v_r be the expected number of violators for a random subset of cardinality r.
Let x_r be the expected number of extreme points for a random subset of cardinality r

$$v_r = x_{r+1}(n-r)/(r+1)$$

Thus for the expected number of violated constraints it holds
$E(|V|) \leq d(n - d\sqrt{n})/(d\sqrt{n}+1) \leq \sqrt{n}$.

Analysis

- Let B be the optimal basis.
- Sampling Lemma: $E(|V|) = \sqrt{m}$.
- Markov: $P(|V| > 2\sqrt{m}) \leq 1/2$.
- In each augmentation set a new element of B enters G (Fact 1).
- Thus there are at most d augmentation steps.
- The expected number of rounds is $2d$.

Clarkson 1 performs:

- Expected 2d linear programs with Clarkson 2 on at most $3d\sqrt{m}$ constraints.
- Expected $O(d^2 m)$ additional arithmetic operations.

Analysis

- Let B be the optimal basis.
- Look after kd iterations of reweighting:
  - $\mu(B) \geq 2^k$, since at least one μ_h has doubled k times.
  - $\mu(B) \leq m e^{k/3}$, since $\mu(V) \leq \mu(H)/(3d)$
- Thus $2^k \leq m e^{k/3}$, i.e. $k \leq \log m + k/(3 \ln 2)$, i.e. $k \leq 2 \log m$.
- Consider H as multiset, i.e. each h has multiplicity μ_h and use sampling lemma.
- Thus the probability of reweighting is at least $1/2$, since $E(V) = d(\mu(H) - r)/(6d^2 + 1) \leq \mu(H)/(6d)$.

Clarkson 2

Input: A set H of linear constraints, objective function c.

Output: Optimal point x^*.

Algorithm: Clarkson2(H)

- Let $r = 6d^2, \mu_h = 1 \forall h \in H$
- Repeat:
  - Choose random $R \in \binom{H}{r}$
  - Compute $x^* =$ Simplex(R).
  - Let $V = \{h \in H \mid x^*$ violates $h\}$
  - If $\mu(V) \leq \mu(H)/(3d)$ then $\mu_h = 2\mu_h$ for all $h \in V$.
- Until $V = \emptyset$
- Return x^*

Integer Linear Programming (without proofs)

$$\begin{array}{llcl} \min & c^T x & & \\ s.t. & Ax & \leq & b \\ & x & \in & \mathbb{Z}^d \end{array}$$

- We denote the dimension of the space by d.
- m denotes the number of inequalities in $Ax \leq b$.
- There is an algorithm (due to Lenstra) that solves a LP with a fixed number of constraints in fixed dimension in $O(s^2)$, where s is the binary encoding length of the input.

IP bases

Theorem (Bell and Scharf):

- If $\min\{c^T x \mid x \in \mathbb{Z}^d, x$ satisfies all $h \in H\}$ is finite and non-empty then there exists a subset $H' \subseteq H$ of size at most $2^d - 1$ such that $\min\{c^T x \mid x \in \mathbb{Z}^d, x$ satisfies all $h \in H'\} = \min\{c^T x \mid x \in \mathbb{Z}^d, x$ satisfies all $h \in H\}$
- If $\min\{c^T x \mid x \in \mathbb{Z}^d, x$ satisfies all $h \in H\}$ is empty then there exists a subset $H' \subseteq H$ of size at most $2^d - 1$ such that $\min\{c^T x \mid x \in \mathbb{Z}^d, x$ satisfies all $h \in H'\}$ is empty.

IP Algorithm

Theorem (Clarkson): An integer program with m constraints each of encoding length at most s can be solved in time $O(m + \log m s^2)$.

LP-type Problems

The Clarkson-algorithm can be applied in the following general framework.

Note that Linear Programming, Integer Programming and Smallest Enclosing Balls fall into this framework

Definition:

Let S be a finite set, O some ordered set, and $\phi : 2^S \mapsto O \cup \{\perp\}$ a function, where $\perp$ is assumed to be the minimum value in $O \cup \{\perp\}$.
(S, ϕ) is called an *LP-type problem*, if ϕ is monotone and local, i.e. if for all $R \subseteq Q \subseteq S$ with $\phi(R) \neq \perp$
(i) $\phi(R) \leq \phi(Q)$, and
(i) $\phi(R) = \phi(Q)$ implies $V(R) = V(Q)$.

LP-type Problems

Definition:

- A basis for $R \subseteq S$ is an inclusion-minimal set $B \subseteq R$ such that $\phi(B) = \phi(R)$.
- The combinatorial dimension of a LP-type problem is the size of the largest basis of any $R \subseteq S$.
- Given a basis B and an element $b \in S$ a basis computation is the computation of the optimal basis of $B \cup \{b\}$.

Theorem:

An LP-type problem of combinatorial dimension d can be solved with an expected number of at most $O(d^2 m + e^{O(\sqrt{(d \log d)})})$ violation tests resp. basis computations.

We showed $O(d^2 m + d^2 \log m$Small-Instances$)$.