Randomized Rounding

- Brief Introduction to Linear Programming and Its Usage in Combinatorial Optimization
- Randomized Rounding for Cut Problems
- Randomized Rounding for Covering Problems
- Randomized Rounding for Satisfiability Problems
- **Randomized Rounding and Semi-definite Programming**

Approximate Sampling and Counting

- ...
**MAX-CUT and MAX-2SAT**

**MAX-CUT**

**Input:** graph $G = (V, E)$, $w : E \rightarrow \mathbb{N}$

**Output:** a cut $(S, \bar{S})$, $S \subset V$, with maximum total weight of edges crossing the cut.

**MAX-2SAT**

**Input:** a 2-CNF formula $\varphi$, $n$ variables, $m$ clauses, clause $j$ is “weighted” with $w_j \in \mathbb{N}$

**Output:** a truth assignment maximizing the total weight of satisfied clauses
QCQP and Strict QCQP

**Definition (Quadratically Constrained Quadratic Program – QCQP)**
Optimize a quadratic function subject to quadratic constraints.

**Definition (Strict QCQP)**
Optimize a quadratic function subject to quadratic constraints. The monomials in the objective function and in the constraints are all of degrees 2 or 0.
MAX-2SAT as a QCQP

Think: \( y_i = 1/0 \) iff \( x_i = \text{TRUE}/\text{FALSE} \)

Example:

\[ \varphi = (\overline{x}_1 \lor x_2) \land (x_3) \land (x_1 \lor \overline{x}_3) \]

\[ \max \quad w_1(1 - y_1(1 - y_2)) + w_2(1 - (1 - y_3)) + w_3(1 - (1 - y_1)y_3) \]

subject to

\[ y_i^2 = y_i, \quad \forall i \]

\[ y_i \in \mathbb{R}, \quad \forall i \]
**MAX-CUT as a Strict QCQP**

Think: \( x_i = 1 \) or \(-1\) iff vertex \( i \in \) or \( \notin S \)

\[
\text{max} \quad \frac{1}{2} \sum_{ij \in E} w_{ij}(1 - x_ix_j)
\]

subject to \( x_i^2 = 1, \ \forall i \in V \quad x_i \in \mathbb{R}, \ \forall i \in V \)
Vector Program

Definition (Vector Program)

Variables: \( n \) vectors \( \mathbf{v}_1, \ldots, \mathbf{v}_n \) in \( \mathbb{R}^n \)

Objective and Constraints: linear in the inner products \( \langle \mathbf{v}_i, \mathbf{v}_j \rangle \)

The general form of a vector program is

\[
\begin{align*}
\text{max} & \quad \sum_{1 \leq i, j \leq n} c_{ij} \langle \mathbf{v}_i, \mathbf{v}_j \rangle \\
\text{subject to} & \quad \sum_{1 \leq i, j \leq n} a_{ij}^{(k)} \langle \mathbf{v}_i, \mathbf{v}_j \rangle = b_k \quad 1 \leq k \leq m \\
& \quad \mathbf{v}_i \in \mathbb{R}^n \quad \forall i \in [n]
\end{align*}
\]
From a Strict QCQP, we easily get a “relaxed” vector program by replacing each variable with a vector, and a product of two variables with the inner product of the corresponding vectors.

\[
\text{max } \frac{1}{2} \sum_{ij \in E} w_{ij} (1 - \langle v_i, v_j \rangle)
\]

subject to \[ \langle v_i, v_i \rangle = 1, \quad \forall i \in V \]
\[ v_i \in \mathbb{R}^n, \quad \forall i \in V \]
Why Vector Programs?

- A vector program (VP) can be solved to within $\pm \epsilon$ of optimality in time polynomial in the input size and $\log(1/\epsilon)$
- Reason: vector program is equivalent to semidefinite program
- After getting a (near) optimal solution $v_1^*, \ldots, v_n^*$ to the vector program, we can (randomly) “round” back to a feasible solution $x^A$ of the original optimization problem.
- Sometime, a problem can be relaxed directly to a semidefinite program (SDP)
- Thus, need to know SDP and its equivalence with VP
Positive Semidefinite Matrices

Definition/Characterization: given a real and symmetric \( n \times n \) matrix \( A \), the following are equivalent

- \( A \) is positive semidefinite
- \( x^T A x \geq 0 \), for all \( x \in \mathbb{R}^n \)
- all eigenvalues of \( A \) are non-negative
- \( A = W^T W \) for some real matrix \( W \) (not necessarily square)
- \( A \) is a nonnegative linear combination of matrices of the type \( xx^T \)
- the determinant of all symmetric minor of \( A \) is non-negative

More notations

- Use \( A \in \mathbb{R}^{n \times n} \) to denote “\( A \) is an \( n \times n \) real matrix”
- Use \( A \succeq 0 \) to denote “\( A \) is positive semidefinite” (PSD)
- Use \( S_n \) to denote the set of all symmetric matrices in \( \mathbb{R}^{n \times n} \)
- For \( C, X \in S_n \), the Frobenius inner product of them is

\[
C \cdot X := \text{tr } C^T X = \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij}.
\]
**Definition (Semidefinite Program – SDP)**

Optimizing a linear function of the $x_{ij}$ subject to linear constraints on them, and subject to $X = (x_{ij}) \succeq 0$

In particular, let $C, A_1, \ldots, A_m \in S_n$, and $b_1, \ldots, b_m \in \mathbb{R}$. The following is a general SDP:

$$\max \quad C \cdot X$$
$$\text{subject to} \quad A_i \cdot X = b_i \quad 1 \leq i \leq m$$
$$X \succeq 0$$

If all $C, A_1, \ldots, A_m$ are diagonal matrices, then the SDP is an LP.
Solving Semidefinite Programs

Theorem

A semidefinite program can be solved to within an additive factor $\epsilon$ of optimality in time polynomial in $n$ and $\log(1/\epsilon)$

Two basic methods:

- Ellipsoid
- Interior point
Vector Program $\equiv$ Semidefinite Program

Vector Program

$$\begin{align*}
\text{max} & \quad \sum_{1 \leq i, j \leq n} c_{ij} \langle v_i, v_j \rangle \\
\text{subject to} & \quad \sum_{1 \leq i, j \leq n} a_{ij}^{(k)} \langle v_i, v_j \rangle = b_k \quad 1 \leq k \leq m \\
& \quad v_i \in \mathbb{R}^n \quad \forall i \in [n]
\end{align*}$$

Semidefinite Program

$$\begin{align*}
\text{max} & \quad C \cdot X \\
\text{subject to} & \quad A_k \cdot X = b_k \quad 1 \leq k \leq m \\
& \quad X \succeq 0
\end{align*}$$

- From (1) to (2), set $x_{ij} = \langle v_i, v_j \rangle$
- From (2) to (1), write $X = W^T W$ (possible since $X$ is PSD), then set $v_i$ to be the $i$th column of $W$
Randomized Rounding for **MAX-CUT**

The Vector Program (i.e. the SDP) for **MAX-CUT**

$$\text{max} \quad \frac{1}{2} \sum_{ij \in E} w_{ij} (1 - \langle v_i, v_j \rangle)$$

subject to

- $\langle v_i, v_i \rangle = 1, \forall i \in V$
- $v_i \in \mathbb{R}^n, \forall i \in V$

**Intuitions:**

- A feasible solution maps each vertex to a point on the $n$-dimensional unit sphere $S_{n-1}$
- Let $\theta_{ij}$ be the angle between $v_i, v_j$, the contribution of edge $ij$ is $\frac{1}{2} (1 - \langle v_i, v_j \rangle) = \frac{1}{2} (1 - \cos \theta_{ij})$
- The wider separated the $v_i, v_j$, the larger the contribution
- A hyperplane (through the origin) will likely separate $v_i, v_j$ if they are widely separated
- Thus, pick a random hyperplane and “use” it as a cut
Randomized Rounding for \textsc{Max-Cut}

1. Let $v_1^*, \ldots, v_n^*$ be a (near) optimal solution to the vector program.

2. Choose a unit vector $r$ uniformly at random from the unit sphere $S_{n-1}$ (think of it as the normal vector of the random hyperplane).

3. Output the cut $(S, \bar{S})$, where

   \[ S = \{ i \in V \mid \langle v_i^*, r \rangle \geq 0 \} \]
   \[ \bar{S} = \{ i \in V \mid \langle v_i^*, r \rangle < 0 \} \]
Analysis

- For any edge \( ij \in E \),

\[
\text{Prob} [ \mathbf{v}_i, \mathbf{v}_j \text{ are separated by } r ] = \frac{\theta_{ij}}{\pi} = \frac{\arccos ( \langle \mathbf{v}_i, \mathbf{v}_j \rangle )}{\pi}
\]

- Expected cut capacity is thus

\[
\sum_{ij \in E} w_{ij} \frac{\arccos ( \langle \mathbf{v}_i, \mathbf{v}_j \rangle )}{\pi}
\]

\[
= \sum_{ij \in E} \left( \frac{\arccos ( \langle \mathbf{v}_i, \mathbf{v}_j \rangle )}{\pi} \right) w_{ij} \left( \frac{1 - \langle \mathbf{v}_i, \mathbf{v}_j \rangle}{2} \right)
\]

\[
\geq \min_{x \in [-1,1]} \left( \frac{\arccos(x)}{\pi} \right) \cdot \text{OPT(Vector Program)}
\]

\[
\geq 0.87856 \cdot \text{MAX-CUT CAPACITY}
\]
How to choose $r$ uniformly on the sphere?

- What do we mean by “uniform on the sphere anyway?”
  - The uniform distribution of a bounded set $B \subset \mathbb{R}^k$ is the distribution whose density is
    \[
    f(x_1, \ldots, x_k) = \begin{cases} 
    \frac{1}{V} & x \in B \\
    0 & \text{otherwise}
    \end{cases}
    \]
    where $V$ is the $k$-dimensional volume (or Lebesgue measure) of $B$.
  
- Consider $S_1$, the 2-dimensional circle. One way to pick $r$ uniformly is to pick $\theta \in [0, 2\pi]$ uniformly at random.

- This is a continuous distribution, which we have not really talked about.
A r.v. $X$ taking on uncountably many possible values is a **continuous random variable** if there exists a function $f : \mathbb{R} \rightarrow \mathbb{R}$, having the property that for every $B \subseteq \mathbb{R}$:

$$\text{Prob}[X \in B] = \int_B f(x)dx$$

$f$ is called the **(probability) density function** (PDF) of $X$. We must have

$$1 = \text{Prob}[X \in (-\infty, \infty)] = \int_{-\infty}^{\infty} f(x)dx.$$ 

The **(cumulative) distribution function** (CDF) $F(\cdot)$ of $X$ is defined by

$$F(a) = \text{Prob}[X \in (-\infty, a)] = \int_{-\infty}^{a} f(x)dx.$$ 

Note that

$$\frac{d}{da} F(a) = f(a)$$
$X$ is said to be *uniformly distributed* on the interval $[\alpha, \beta]$ if its density is

$$f(x) = \begin{cases} \frac{1}{\beta - \alpha} & \text{if } x \in [\alpha, \beta] \\ 0 & \text{otherwise} \end{cases}$$

As $F(a) = \int_{-\infty}^{a} f(x) dx$, we get

$$F'(a) = \begin{cases} 0 & a < \alpha \\ \frac{a - \alpha}{\beta - \alpha} & a \in [\alpha, \beta] \\ 1 & a > \beta \end{cases}$$
PTCF: Continuous Unif. Dist., Some Density Plots
$X$ is said to be **exponentially distributed** with parameter $\lambda$ if its density is

$$f(x) = \begin{cases} \lambda e^{-\lambda x} & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases}$$

Its cdf $F$ is

$$F(a) = \int_{-\infty}^{a} f(x) dx = 1 - e^{-\lambda a}, \ a \geq 0.$$
PTCF: Exponential Dist., Some Plots

Densities

Distributions

\( \lambda = 0.5 \quad \lambda = 1.0 \quad \lambda = 1.5 \)
A continuous r.v. $X$ is *normally distributed* with parameters $\mu$ and $\sigma^2$ if the density of $X$ is given by

$$f(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \ x \in \mathbb{R}.$$ 

Normal variables are also called *Gaussian* variables.

If $X$ is normally distributed with parameters $\mu$ and $\sigma^2$, then $Y = \alpha X + \beta$ is normally distributed with parameters $\alpha \mu + \beta$ and $(\alpha \sigma)^2$.

When $\mu = 0$ and $\sigma^2 = 1$, $X$ is said to have *standard normal* distribution.
PTCF: Normal Distribution, Some Plots

Densities

Distributions
PTCF: Continuous Distribution Random Number Generation

- Uniform distribution: discretize it, then use some *pseudo-random number generator*
- Let's assume we can generate a uniform number $X \in [0, 1)$.

Question:
- How to generate $Y \in \text{Normal}(\mu, \sigma)$?
- It is actually sufficient to generate $Y \in \text{Normal}(0, 1)$
- How to generate a point on an $n$-sphere uniformly at random?
PTCF: Normal Distribution Random Number Generator

The Polar Method (for Normal(0, 1))

1. Generate $V_1, V_2 \in [-1, 1]$ uniformly
2. $S = V_1^2 + V_2^2$
3. If $S \geq 1$, go back to step 1
4. Set $X_1 = V_1 \sqrt{-\frac{2 \ln S}{S}}$ and $X_2 = V_2 \sqrt{-\frac{2 \ln S}{S}}$
   Then, $X_1$ and $X_2$ are independent standard normal variables

For Normal($\mu, \sigma$)

1. Let $X$ be a standard normal variable
2. Then, $Y = \mu + \sigma X$ is Normal($\mu, \sigma$)
Generate $X_1, \ldots, X_n$ independently from Normal$(0, 1)$

Let $r = (r_1, \ldots, r_n)$ be defined by

$$r_i = \frac{X_i}{\sqrt{X_1^2 + \cdots + X_n^2}}$$

The joint density of the $X_i$ only depends on $\sqrt{X_1^2 + \cdots + X_n^2}$, so the distribution is spherically symmetric, and thus its projection on to the sphere (i.e. $r$) is uniformly distributed on the surface of the sphere!