22.1 Correlation Clustering

- Input:
  - Graph $G = (V, E)$
  - Weight functions $w^- : E \to \mathbb{R}^+$ and $w^+ : E \to \mathbb{R}^+$
- Feasible: Partition $\mathcal{S} = S_1, S_2, \ldots, S_n$ of V. Given $\mathcal{S}$, let $\delta(\mathcal{S})$ be the edges between different parts of the partition, and let $E(\mathcal{S})$ be the edges with both endpoints in same part of partition.
- Objective: $\max \sum_{\{i,j\} \in E(\mathcal{S})} w^+(i,j) + \sum_{\{i,j\} \in \delta(\mathcal{S})} w^-(i,j)$

Simple 1/2-approximation: if we set $\mathcal{S} = \{V\}$ then objective is $\sum_{e \in E} w^+(e)$, while if we set $\mathcal{S} = \{\{i\} : i \in V\}$ objective is $\sum_{e \in E} w^-(e)$. Since for any $\mathcal{S}$ the objective is at most $\sum_{e \in E} w^-(e) + \sum_{e \in E} w^+(e)$, taking the best of the two is a 1/2-approximation.

To do better we will write an SDP. Recall that e_k is the kth standard basis vector: the vector with a 1 in the kth coordinate and a 0 everywhere else. Then the following vector program is an exact formulation of Correlation Clustering:

$$
\begin{aligned}
\max \quad & \sum_{\{i,j\} \in E} \big(w^+(i,j)(v_i \cdot v_j) + w^-(i,j)(1 - v_i \cdot v_j) \big) & \\
\text{s.t.} \quad & v_i \in \{e_1, e_2, \ldots, e_n\} & \forall i \in V
\end{aligned}
$$

This is because if i and j get the same basis vector e_k we can think of them as being in the same part of the partition (the kth part), in which case their dot product is 1 and the objective get $w^+(i,j)$, while if i and j get different basis vectors then they are in different parts so their dot product is 0 and hence the objective get $w^-(i,j)$.

We cannot solve this vector program, but we can relax it to an SDP:

$$
\begin{aligned}
\max \quad & \sum_{\{i,j\} \in E} \big(w^+(i,j)(v_i \cdot v_j) + w^-(i,j)(1 - v_i \cdot v_j) \big) & \\
\text{s.t.} \quad & v_i \cdot v_i = 1 & \forall i \in V \\
& v_i \cdot v_j \geq 0 & \forall i, j \in V \\
& v_i \in \mathbb{R}^n & \forall i \in V
\end{aligned}
$$

We will round this SDP using random hyperplanes, as we did with Max-Cut. But instead of using a single random hyperplane, we will use two independent random hyperplanes. Slightly more formally, given a solution to the SDP, we will construct a partition into four parts by choosing random unit vectors r_1 and r_2 and defining the following sets:

$$\begin{aligned}
R_1 &= \{i \in V : r_1 \cdot v_i \geq 0,\ r_2 \cdot v_i \geq 0\} \\
R_2 &= \{i \in V : r_1 \cdot v_i \geq 0,\ r_2 \cdot v_i < 0\} \\
R_3 &= \{i \in V : r_1 \cdot v_i < 0,\ r_2 \cdot v_i \geq 0\} \\
R_4 &= \{i \in V : r_1 \cdot v_i < 0,\ r_2 \cdot v_i < 0\}.
\end{aligned}$$

We let $\mathcal{S} = \{R_1, R_2, R_3, R_4\}$.

Let X_{ij} be a random variable which is 1 if vertices i and j end up in the same cluster. We saw in the last lecture that the probability that a single random hyperplane separates i and j is $\frac{\theta_{ij}}{\pi} = \frac{\arccos(v_i \cdot v_j)}{\pi}$. Hence $\mathbf{E}[X_{ij}] = (1 - \frac{1}{\pi}\arccos(v_i \cdot v_j))^2$. Let W be the value of the objective function for our partition $\mathcal{S}$. Then

$$W = \sum_{\{i,j\} \in E} \left(w^+(i,j) X_{ij} + w^-(i,j)(1 - X_{ij}) \right),$$

an so by linearity of expectations

$$\begin{aligned}
\mathbf{E}[W] &= \sum_{\{i,j\} \in E} \left(w^+(i,j)\mathbf{E}[X_{ij}] + w^-(i,j)(1 - \mathbf{E}[X_{ij}]) \right) \\
&= \sum_{\{i,j\} \in E} \left(w^+(i,j) \left(1 - \frac{1}{\pi}\theta_{ij}\right)^2 + w^-(i,j) \left(1 - \left(1 - \frac{1}{\pi}\theta_{ij}\right)^2\right)\right)
\end{aligned}$$

It turns out due to trig/calculus that $(1 - \frac{\theta_{ij}}{\pi})^2 \geq \frac{3}{4}\cos(\theta_{ij})$ and that $1 - (1 - \frac{\theta_{ij}}{\pi})^2 \geq \frac{3}{4}(1 - \cos(\theta_{ij}))$, as long as $\theta_{ij} \leq \pi/2$. But $\theta_{ij} \leq \pi/2$ because $v_i \cdot v_j \geq 0$. Hence we have that

$$\begin{aligned}
\mathbf{E}[W] &\geq \sum_{\{i,j\} \in E} \left(w^+(i,j) \left(\frac{3}{4}\cos\theta_{ij}\right) + w^-(i,j) \left(\frac{3}{4}(1 - \cos\theta_{ij})\right)\right) \\
&= \frac{3}{4} \sum_{\{i,j\} \in E} \left(w^+(i,j)(v_i \cdot v_j) + w^-(i,j)(1 - v_i \cdot v_j) \right) \\
&= \frac{3}{4} \cdot OPT_{SDP}
\end{aligned}$$

Thus this is a 3/4-approximation. This algorithm and analysis is due to Swamy [SODA 2004].

22.2 Max-2SAT

Max-2SAT is the following problem:

- Input:
  - n variable $x_1, \ldots, x_n$
  - m CNF clauses $C_1, \ldots, C_m$, each of which has exactly two literals
- Feasible: assignment of T/F to variables
- Objective: maximize number of satisfied constraints

Writing a strict quadratic program for Max-2SAT is actually a bit tricky. To see this, suppose that we have a variable $y_i \in \{-1, 1\}$ for each input variable, where assigning $y_i = -1$ corresponds to setting x_i to T and assigning $y_i = 1$ corresponds to setting x_i to F. Consider a clause of the form $x_i \vee \bar{x_j}$. Then assigning $y_i = -1$ and $y_j = 1$ corresponds to a satisfying assignment, but assigning $y_i = 1$ and $y_j = -1$ is not a satisfying assignment. But in a strict quadratic program, where we can only use terms like $y_i y_j$, there is no way of distinguishing between these assignments!

So we can't just think of $y_i = 1$ as a necessarily true or false. Instead, we'll add a new "dummy" variable $y_T \in \{-1, 1\}$, and whatever value y_T gets is what we define as T. This means that for $x_i \vee x_j$ and an assignment y_T, y_i, y_j, if the assignment satisfies the clause

$$\frac{3 + y_i y_T + y_j y_T - y_i y_j}{4} = 1$$

and otherwise it equals 0. If the clause has negated variables (e.g. $x_i \vee \bar{x}_j$), then we just use the same formula but with negated variables corresponding to negative (integer) variables. So for this example, we would negate y_j to get

$$\frac{3 + y_i y_T - y_j y_T + y_i y_j}{4}.$$

This lets us write the following strict quadratic program:

$$\begin{aligned}
\max \quad & \sum_{\text{clauses } x_i \vee x_j} \frac{3 + y_i y_T + y_j y_T - y_i y_j}{4} + \sum_{\text{clauses } x_i \vee \bar{x}_j} \frac{3 + y_i y_T - y_j y_T + y_i y_j}{4} \\
& + \sum_{\text{clauses } \bar{x}_i \vee x_j} \frac{3 - y_i y_T + y_j y_T + y_i y_j}{4} + \sum_{\text{clauses } \bar{x}_i \vee \bar{x}_j} \frac{3 - y_i y_T - y_j y_T + y_i y_j}{4} \\
\text{s.t.} \quad & y_i \in \{-1, 1\} \qquad \forall i \in V \\
& y_T \in \{-1, 1\}
\end{aligned}$$

When we relax this to an SDP, we get the following:

$$
\begin{aligned}
\max \quad & \sum_{\text{clauses } x_i \vee x_j} \frac{3 + v_i \cdot v_T + v_j \cdot v_T - v_i \cdot v_j}{4} + \sum_{\text{clauses } x_i \vee \bar{x}_j} \frac{3 + v_i \cdot v_T - v_j \cdot v_T + v_i \cdot v_j}{4} \\
& + \sum_{\text{clauses } \bar{x}_i \vee x_j} \frac{3 - v_i \cdot v_T + v_j \cdot v_T + v_i \cdot v_j}{4} + \sum_{\text{clauses } \bar{x}_i \vee \bar{x}_j} \frac{3 - v_i \cdot v_T - v_j \cdot v_T + v_i \cdot v_j}{4} \\
\text{s.t.} \quad & v_i \cdot v_i = 1 \qquad \forall i \in V \\
& v_i \in \mathbb{R}^n \qquad \forall i \in V \\
& v_T \cdot v_T = 1 \\
& v_T \in \mathbb{R}^n
\end{aligned}
$$

We can round this using random hyperplane rounding (again): we choose a random r, and we set to true every x_i with $sign(v_i \cdot r) = sign(v_T \cdot r)$ and set to false all x_i with $sign(v_i \cdot r) \neq sign(v_T \cdot r)$. And if x_i is set to true then we set y_i to -1 and if we set x_i to false then we set y_i to 1.

To analyze this, let's rewrite the term for each clause. For simplicity, we'll just consider the first type of clauses $x_i \vee x_j$ (the other cases are similar). Then we can rewrite $\frac{3+v_i \cdot v_T + v_j \cdot v_T - v_i \cdot v_j}{4}$ and $\frac{1}{4}((1 + v_i \cdot v_T) + (1 + v_j \cdot v_T) + (1 - v_i \cdot v_j))$. Then all terms look like $(1 \pm v \cdot u)$ for vectors u and v. We'll analyze each of these types separately. Recall that $\alpha_{GW} = \inf_{0 \leq \theta \leq \pi} \frac{2\theta}{\pi(1-\cos\theta)}$.

Consider a term $1 - v_i \cdot v_j$ (where possibly either i or j is T). Then the contribution to the SDP of this term is $1 - v_i \cdot v_j = 1 - \cos\theta_{ij}$. On the other hand, the probability that i and j are on different sides of the hyperplane is θ_{ij}/π. Hence the expected value of this term in the rounded solution is $\mathbf{E}[1 - y_i y_j] = 2\frac{\theta_{ij}}{\pi} \geq \alpha_{GW}(1 - v_i \cdot v_j)$.

Similarly, consider a term $1 + v_i \cdot v_j$. Then the contribution to the SDP is $1 + v_i \cdot v_j = 1 + \cos\theta_{ij}$. In the rounded solution, the expected value is $\mathbf{E}[1 + y_i y_j] = 2(1 - \frac{\theta_{ij}}{\pi})$. Hence the ratio between the integral contribution and the SDP contribution is $\frac{2(1-\theta_{ij}/\pi)}{1+\cos\theta_{ij}} = \frac{2(\pi - \theta_{ij})}{\pi(1+\cos\theta_{ij})}$. If we set $\theta' = \pi - \theta_{ij}$, then by basic trig we get that this is equal to $\frac{2\theta'}{\pi(1-\cos\theta')} \geq \alpha_{GW}$ (since $\cos(\pi - \theta) = -\cos(\theta)$).

Since for every term the expected contribution to our integral solution is at least α_{GW} times the expected contribution to the SDP solution, by linearity of expectations this gives us an α_{GW}-approximation.