# 1 May 9th, 2019

## 1.1 MAX-2SAT

For each variable $x_i$, introduce a baraible $y_i$ which is either +1 or -1. Also introduce an extra variable $y_0$, whihc is also either +1 or -1.

### Intuition

Variable $x_i$ is true of $y_i = y_0$ and $x_1$ is false if $y_i \neq y_0$. Consider the following clauses:

$$v(x_i) \to \frac{1}{2}\left(1 + y_i y_0\right).$$

As such, we can write the objective function as:

$$\max \frac{1}{2}\sum_i + \frac{1}{2}(1 + y_i y_0).$$

However, we need to consider the negation. Note that if it is we have a $\overline{x}_j$ clause, then we we would have:

$$v(\overline{x}_j) \to \frac{1}{2}\left(1 - y_i y_0\right).$$

As such, we have:

$$\max \sum v(\text{clause}_i).$$

Since the objective function contains the product, then we can change this into a vector program, since it can correspond to the dot product of two vectors. This is why we introduce $y_0$.

### Checking Validity

$$\begin{aligned} v(x_i \vee x_j) = 1 - v(\overline{x}_i)v(\overline{x}_j) &= 1 - \frac{1}{2}(1 - y_i y_0)\frac{1}{2}(1 - y_j y_0) \\ &= 1 - \frac{1}{4}\left[1 - y_i y_0 - y_j y_0 + y_i y_j y_0^2\right]. \end{aligned}$$

Note that $y_0^2 = 1$ :

$$\begin{aligned} &\frac{3}{4} + \frac{1}{4}y_i y_0 + \frac{1}{4}y_j y_0 - \frac{1}{4}y_i y_j \\ &= \frac{(1 + y_i y_0) + (1 + y_j y_0) + (1 - y_i y_j)}{4} = \frac{1}{4}(1. \end{aligned}$$

Which is of the dot product. TODO: compute $v(\overline{x}_i \vee x_j)$ and other combinations As such, the objective function is of form:

$$\max \sum_{0 \leq i < j \leq n} \left[a_{ij}(1 + y_i y_j) + b_{ij}(1 - y_i y_j)\right],$$

with some coefficient $a_{ij}$, $b_{ij}$. Note that $i < j$ because of $y_0$

**Constraints**

Since $x_i \in \{0, 1\}$, we have

$$y_i^2 = +1, \text{ for } 0 \leq i \leq n.$$

Note that the MAX-2SAT is the special case of this problem.

## 1.2 Vector Program for MAX-2SAT

Relax to get vector program:

$$\max \sum_{0 \leq i < j \leq n} a_{ij}(1 + v_i \cdot v_j) + b_{ij}(1 - v_i \cdot v_j).$$

Subject to:

$$v_i \cdot v_i = 1$$
$$v_i \in \mathbb{R}^{n+1}.$$

Assume we have solved this exactly, and now we will round it. We will now round it. First lets make some modifications:

$$W = \max 2 \sum_{0 \leq i < j \leq n} a_{ij} \frac{1 + y_i y_j}{2} + b_{ij} \frac{1 - y_i y_j}{2}.$$

$$E(W) = 2 \sum a_{ij} \Pr(y_i = y_j) + b_{ij} \Pr(y_i \neq y_j).$$

This is the expected number of clauses satisfied by our algorithm. Comparing this to our vector program, we have:

$$\max 2 \sum_{0 \leq i < j \leq n} a_{ij} \frac{1 + v_i \cdot v_j}{2} + b_{ij} \frac{1 - v_i \cdot v_j}{2}.$$

Note that we want our randomized rounding to have the following properties:

- If two vectors $v_i$ and $v_j$ are pointing in the same direction, $\Pr(y_i = y_j)$ should be close to 1.
- If two vectors are in opposite direction, then $\Pr(y_i \neq y_j)$ is close to 1.

This is similar to the max-cut analysis, in which we take a random hyper plane to separate. Last time we analyzed that:

$$\Pr(y_i \neq y_j) = \frac{\theta}{\pi}.$$

## 1.3 Approximation Factor

For this, we have to show:

$$\frac{\Pr(y_i = y)j}{\frac{(1+v_i \cdot v_j)}{2}} \geq 0.878 \text{ and } \frac{\Pr(y_i \neq y)j}{\frac{(1 v_i \cdot v_j)}{2}} \geq 0.878.$$

Which will give us an approximation factor of 0.878. As we know, we have:

$$\frac{1 - v_i \cdot v_j}{2} = \frac{1 - \cos(\theta_{ij})}{2} = \ldots.$$

Next, we have:

$$\Pr(y_i = y_j) = 1 - \frac{\theta_{ij}}{2}.$$

Comparing with $\frac{1+\cos(\theta_{ij})}{2}$. What we know is that:

$$\frac{\frac{\theta}{\pi}}{\frac{1-\cos(\theta)}{2}} \geq 0.878.$$

Which is true for any $\theta$. Replacing $\theta$ by $\pi - \theta$, we get the desired inequality.

**Remark 1.1** — This is very similar to the max-cut, but the difference is we have to introduce a new variable $y_0$.

**Remark 1.2** — When we used integer programming for the max-cut, we got nothing, but for MAX-SAT, we can get good results too, getting an approx ratio of 0.75.

## 1.4 MAX-2SAT But with Integer Linear Programming

For each boolean variable $x_i$, introduce a 0/1 variable $y_i$. For clause $j$, introduce 0/1 variable $z_j = 1$ if clause is true. If our clauses are:

$$(x_1 \vee \overline{x}_2 \vee x_3$$

$$\overline{x}_1 \vee x_2 \vee \overline{x}_3.$$

Our ILP would be is:

$$\max_j z_j$$

Subject to:

$$z_1 = \leq y_1 + (1 - y_2) + y_3 z_2 \leq (1 - y_1) + y_2 + (1 - y_3)$$
$$z_j \in \{0, 1\}$$
$$y_i \in \{0, 1\}.$$

Relaxing would give us would be is:

$$\max_j z_j$$

Subject to:

$$z_1 = \leq y_1 + (1 - y_2) + y_3 z_2 \leq (1 - y_1) + y_2 + (1 - y_3)$$
$$0 \leq z_j \leq 1$$
$$0 \leq y_i \leq 1.$$

Randomized Rounding: Each variable $x_i$ is set to true independently with probability $y_i$ (after solving relaxed LP). Otherwise set to false. Consider a clause with $k$ literals, $x_1 \vee x_2 \vee \ldots v_k$ (no negation but analysis would hold still) We have:

$$\Pr(\text{Clause is satisfied}) = 1 - (1 - y_1)(1 - y_2) \ldots (1 - y_k).$$

We know that: $y_1 + y_2 + \ldots + y_j \geq z_j$. In the worse case, we have :

$$\Pr(\text{Clause is satisfied}) \geq 1 - \left(1 - \frac{z_j}{k}\right)^k \geq 1 - \left(1 - \frac{1}{k}\right)^k z_j.$$

For high $k$ this approaches an approximation factor of $1 - \frac{1}{e}$. For MAX-2SAT, this is $\frac{3}{4}$.

**Final Coverage**

- steiner forest
- multiway cut
- semi-definite programming
- well-characterized problems