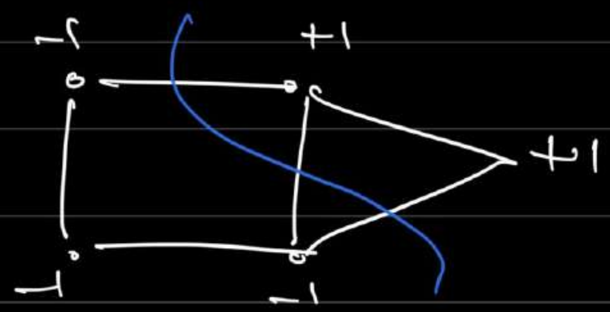
Maxcut problem -

diff nodes → cut
Same nodes → NO cut



our objective is to maximise this cut out of all n nodes.

we have to general n-node graph and calculate cut values return the max of it

$x_i = +1$ or -1

$x = (x_1\ x_2\ x_3\ - - -\ x_n)$

2^n possibilites to generate a graph of n nodes

$\boxed{\text{if } n \longrightarrow \infty \quad \text{NP Hard problem}}$

$x_i x_j$ if -1 cut

if $+1$ No cut

Rewrite

$\frac{1 - x_i x_j}{2}$ if 1 cut

if 0 No cut

Objective :- $\max \sum \left(\frac{1 - x_i x_j}{2} \right)$

if w is the weight of edge

$$\max \left(w_{ij} \left(\frac{1 - x_i x_j}{2} \right) \right)$$

Rewrite in matrix notation

Consider matrix $X = x x^T$

let $x = n \times 1$

$= [x_1\ x_2\ \cdots\ x_n]$

so X_{ij} values are all $x_i x_j$

$$\Rightarrow \max \left[w_{ij} \left(\frac{1 - x_i x_j}{2} \right) \right] \Rightarrow \max \left[w_{ij} \left(\frac{1 - X_{ij}}{2} \right) \right]$$

New objective

$$\max\left[\omega_{ij}\left(\frac{1-X_{ij}}{2}\right)\right]$$

then this X has properties

① $X_{ii} = 1$ - - - → $\begin{bmatrix} 1 \times 1 = 1 \\ -1 \times -1 = 1 \end{bmatrix}$

② $X \succeq 0$

↳ positive semi definite matrix. as it can be written as $X = xx^T$

③ $\text{Rank}(X) = 1$ → as it is produced by single vector

④ X is symmetric

This is still Hard to solve as $x \in (-1, +1)$

This Binary input is making it Rank 1

In General, low rank matrix is easy to solve compared to high rank.

But as Rank 1 is NP-Hard we try to find next low Rank solution.

we know $1 \leq$ Rank of matrix $< \infty$

New objective -

$$\text{maximise}\left(\sum \omega_{ij} \frac{[1 - X_{ij}]}{2}\right)$$

$X \succeq 0$

Rank > 1

$X_{ii} = 1$

X is symmetric

$\Rightarrow \quad X = VV^T$

$V = n \times K$ matrix

instead of $x = n \times 1$ size

we are taking a vector of points

$\Rightarrow V = n \times K$ size

this is called SDP relaxation. There is a method proposed long before to solve SDP problem.

In this paper they proposed new method of solving this SDP which is EP-SDP

we add some entropy to our objective fonction. This entropy pushes the function to become low-Rank (Not certainly Rank 1)

So the flow is

Rank 1 $\longrightarrow$ High Rank $\longrightarrow$ push close to Rank 1

Entropy fonctions -

1. Tsallis entropy - $\frac{1}{1-\alpha}\left[\frac{tr(X^{\alpha})}{(tr X)^{\alpha}} - 1\right]$

2. Renyi entropy - $\frac{\alpha}{1-\alpha} \frac{X^{\alpha-1}}{tr X^{\alpha}}\left(I - \frac{X}{tr X}\right) V$

3. Neumann entropy - $\frac{tr X I - X}{(tr X)^2}\left[I + \log \frac{X}{tr X}\right] V$

Here we will use Tsallis entropy.

New objective function

$$\max \sum w_{ij}\left(\frac{1-X_{ij}}{2}\right) - \lambda R(X)$$

λ - controlls strength of entropy
$R(X)$ - entropy function

① we use gradiant descent to solve this Objective.

② The to the result we apply GW Rounding and convert back to +1 and -1

① Gradiant descent

$$V \longrightarrow V + \eta(\nabla)$$

η is step value
∇ is differentiate of V

In our objective $X = VV^T$
so we try to find Best V

Algorithm -

choose rank k

Initialise V randomly

Normalise each row $\quad v_i \leftarrow \frac{v_i}{\|v_i\|}$

set penalty $\lambda \leftarrow \lambda_0$

for each penalty stage $t = 1, 2 - T$

for each iteration $m = 1, 2 -- M$

compute gradiant

[$f_{un} \leftarrow f_{un} + \lambda(\nabla)$]

diff obj + diff entropy

$$V \leftarrow V - \eta\,(g_f + \lambda g_R)$$

obj diff + entropy diff

$$\lambda \leftarrow \rho \lambda_0$$

We will get V, apply GW Rounding to Convert to +1 and -1

② GW Rounding -

for a value k

if A·k and k has same sign → +1

has diff sign → -1

K is an arbitary value

A is our v matrix

Using this gradiant based entropy penalty proved to give us much faster solution than regual SDP sowing.

Entropy Penalised

Semi-Definite Programming

19-02-2026