Another Example

→ Boolean formula in CNF (conjunctive normal form) — also called P-O-S (product of sum)

Each clause is a disjunction of literals. and the formula is a conjunction of clauses. Another name for CNF is product of sums.

Clauses:
$C_1 = x_1 \vee \overline{x_2} \vee x_4$ (∨ : sum)
$C_2 = x_2 \vee x_3$
$C_3 = x_3 \vee \overline{x_4}$

formula $\phi = C_1 \wedge C_2 \wedge C_3$ (∧ : Product)

→ If 'm' clauses in a formula, there exists a truth assignment, s.t. at least $\frac{m}{2}$ clauses are satisfied.

→ Truth assignment :

x_1	T
x_2	F
⋮	T
⋮	⋮
x_n	T

So a clause will ~~not be~~ evaluate

Teacher's Signature

to true if one of the variables in pure form is true OR if one of the vars in complemented form is false.
i.e.
If one the inputs is true, whole clause is true.

→ So, how do we prepare an assignment that satisfies half the clauses?
we randomly assign
Truth assignment uniformly at random

every x_i { T — 1/2 (prob.)
F — 1/2 (prob.)

→ Why does it satisfy half the clauses.

$$P(C_1 = \text{True}) = 1 - \frac{1}{8} = 1 - \frac{1}{2^3} \quad (1 - 2^{-k})$$

k is no. of variables in a clause C_i

(based on no. of variables in the clause)

$$P(C_1 = T) \geq \frac{1}{2}$$ (because every clause will have at least one variable in it)

$$X_1 = \begin{cases} T & \text{if } C_1 \text{ is True} \\ 0 & \text{o/w} \end{cases}$$

$$E[X_i] \geq \frac{1}{2}$$

$X = \sum X_i$ (X will count clauses that are true)

Teacher's Signature

So, $E[X] \geq \frac{m}{2}$

So, $E[X] = m(1 - 2^{-k})$

$\because k \geq 1$

$E[X] = m/2$ } holds irrespective of wether the formula is satisfiable or not

X — X

Lecture 9 : 05th February 2021

Recap : **CNF Satisfiability** – we aren't interested if there is a satisfying truth assignment for the variables that evaluates ϕ (formula) to true, but we are interested in seeing how many clauses can we satisfy.

Consider each clause as a constraint and the formula as a set of constraints

– we can also put weights on the clauses that (we want to satisfy those constraints that have a higher weight)

Teacher's Signature

Now our problem is to maximise no. of clauses that can be satisfied. — this problem is called MAXSAT.

i.e. can we find an assignment that will satisfy largest possible no. of clauses

→ MAXSAT is also NP-hard indicating that no good polynomial sol^n exists.
→ So, we use randomisation.

Define for an instance I, $m^*(I)$ ($I \xrightarrow{CNF}$ formula) to be the maximum number of clauses that can be satisfied in I.

Let $m^A(I)$ be the number of expected clauses that can be satisfied by an (randomised) algorithm A.

So, the ratio $\frac{m^A(I)}{m^*(I)}$ = performance ratio of algo A.

So this is how many algo A can satisfy irrespective of how you get to the best possible soln.

how many is best possible

$\therefore\ m^A(I) \leq m^*(I)$

$\frac{m^A(I)}{m^*(I)} \leq 1$

the closer the ratio is to 1 the better the algorithm

→ The previous approach (independent VAR) gives $\frac{1}{2}$ as the ratio. or at least 1/2

Previous approach ratio is $1-2^{-k}$

$k = \min$ # variables in a clause

→ There are instances where one can satisfy only 1/2 of the clauses.

eg. C_1 – satisfiable

C_2 – make it unsatisfiable

Task 1: Study An approach that does better than 1/2.

Technique: LP Rounding (Linear Programming R.)

ILP: Within LP, there is a small subset called as integer linear program – where the variable of LP take only integer values.

(Solving ILP is more difficult than solving a simple LP)

Therefore, LP rounding relaxes some of the constraints of the ILP in a step called LP Relaxation to convert the ILP to a simple LP.

(This has both adv. & disadv.)

Teacher's Signature

LP = Ellipsoid algo, interior point, Indian karmaka

SHREE
DATE: / /
PAGE NO.:

Adv is that the relaxed LP can be solved in polynomial time and an optimal solution can be obtained. To satisfy the integrality constraints, we can round the solution from LP.

Disad - while rounding we may lose some quality but that is inevitable.
that is we won't get optimal sol^n but something close to it

→ So now, we have to take our MAXSAT problem and write it as a ILP.

→ The input is a bunch of clauses that are put together using the AND operator. (this isn't a constraint or an objective function in the sense of integer linear programs)

So, first we have to convert input for MAXSAT to ILP. For that, we will take each clause and write it as a constraint

Let z_i be an indicator variable defined to indicate whether C_i is satisfied or not.

C_i	z_i
T	1
F	0

Teacher's Signature

So, we want to **maximise** $\sum_i z_i$

$i = 1$ to m

i ↓ index of clause

objective function

For each boolean variable x_j, we define an indicator variable y_j.

LP variable

x_j	y_j
F	0
T	1

Since variables can appear in either pure or complemented form, we separate them as follows

eg. $C_1 = x_1 \vee \overline{x_2} \equiv y_1 + (1 - y_2)$

x_1	x_2	C_1	$y_1 + (1-y_2)$ LP Value	z_1
T	T	T	1	1
T	F	T	2	1
F	T	F	0	0
F	F	T	1	1

So, to find constraints in the LP world we will look at :

- we want clauses to be satisfied
- so we see what happens to values in LP world when

Teacher's Signature

C_1, x, x_2 take their values in boolean World.

Looking at the table, we can come up with constraints in our 2 LP variables y & z →

So we want to satisfy this constraint:

$$y_1 + (1 - y_2) \geq z_1$$

for the clause $x_1 \vee \overline{x_2}$

$$\Rightarrow \quad x_1 \vee \overline{x_2} \equiv \boxed{y_1 + (1 - y_2) \geq z_1}$$

⇒ whenever z_1 is 1 we satisfy the clause

Therefore,

let C_{i+} be indices of variables in pure form in C_i

let C_{i-} be indices of variables in complemented form in C_i

Now, clause C_i is satisfied if it holds that for each i

$$\sum_{j \text{ in } C_{i+}} y_j + \sum_{j \text{ in } C_{i-}} (1 - y_j) \geq z_i$$

Teacher's Signature

Hence, the entire ILP is :

Maximise $\sum_i z_i$ subject to

$$\sum_{j \text{ in } C_{i+}} y_j + \sum_{j \text{ in } C_{i-}} (1 - y_j) \geq z_i \text{ for all } i$$

where

y_j, z_i in $\{0, 1\}$ for all i & j

Example :

$C_1 = x_1 \vee \overline{x_2} \vee x_4$

$C_2 = x_2 \vee x_3 \vee \overline{x_4}$

$C_3 = \overline{x_1} \vee x_3$

$C_4 = \overline{x_3} \vee \overline{x_4}$

C_1 $y_1 + 1 - y_2 + y_4$

C_2 $y_2 + y_3 + 1 - y_4$

C_3 $1 - y_1 + y_3$

C_4 $2 - y_3 - y_4$

~~x_1 x_2 x_3 x_4 C_1 z_i~~

$y > 0$ was not used in constraints earlier?

→ we want variables y & z to work in unison to make sure objective func works

→ so $y > 0$, values from z are left unconstrained

→ so we have to link the y's and zs

Teacher's Signature

so that when we get values for y_s & z_s, we know what clauses that are satisfied.

Objective func^n = Maximise $\sum_{i=1}^{4} z_i$

Constraints =

① subject to C_1 =

x_1	$\overline{x_2}$	x_4	C_1	z_1	LP-Value
T	T	T	T	1	2
T	T	F	T	1	1
T	F	T	T	1	3
T	F	F	T	1	2
F	T	T	T	1	1
F	T	F	F	0	0
F	F	T	T	1	2
F	F	F	T	1	1

$y_1 + y_4 + 1 - y_2 \geq z_1$

② C_2 : $y_2 + y_3 + 1 - y_4 \geq z_2$

③ C_3 = $1 - y_1 + y_3 \geq z_3$

④ C_4 = $2 - y_3 - y_4 \geq z_4$

and y_s & $z_s \in \{0, 1\}$

Teacher's Signature

SHREE
DATE: / /
PAGE NO.:

RELAXATION : (ILP to LP)

→ Let us allow LP-variables (y_j & z_i) to take any real value b/w 0 & 1.

~~So, we will introduce new variables to navigate across MAXSAT & LP (not ILP)~~ and then find solution :

So, we converted ILP to LP and found u_j & v_i as soln :

		soln	
for y	~~use~~	u	$u_1, u_2 \dots u_n$
for z	~~use~~	v	$v_1, v_2 \dots v_m$

These u_j and v_i are values for the best solun to the relaxed LP.

So, ILP → (y_j, $z_i \in \{0,1\}$)

↓

LP → y_j, $z_i \in [0,1]$

↳ solutn → u_j; v_i

→ and

best soln ← { $\sum_i v_i$ is the VALUE of the obj. f^n in the relaxed LP

and hence is the upper bound on the number of clauses that can be satisfied.

This stands in the ILP as well.

Why ?

Teacher's Signature

Here we are converting LP soln to ILP soln. we can take v_i as soln to ILP also

Because —

$$\sum_i z_i \leq \sum_i v_i$$

as v_i can takes values like 0·1, 0·2 -- z_i will remain zero for such values.

But,

→ The values of v_j are not integral, so they do not yet correspond to T/F in a truth assignment.

So, we still do not have the truth assignment — even after solving LP.

Now, we apply RANDOMISED ROUNDING.

Set y_j (variable for ILP) to 1 with probability u_j

$$y_j = \begin{cases} 1 & \text{prob } u_j \\ 0 & \text{prob } (1-u_j) \end{cases}$$

$$x_j = \begin{cases} T \text{ if } y_j = 1 \\ F \text{ if } y_j = 0 \end{cases}$$

But, when we do this rounding, some of the z variables / clauses may not be having value >0 anymore in LP sense so those will be set to zero.

This means that we will put values of z_i (for after rounding) by looking at constraints

eg. $y_1 + 1 - y_2 \geq z_1$

if LHS is at least 1
we put $z_1 = 1 \Rightarrow C_1 = T$

if LHS < 1
$z_i = 0 \Rightarrow C_1 = F$

So, we do not look for $v_i \rightarrow z_i$ but only $u_j \rightarrow y_j$ when rounding

By doing so, our values of C_i are in line with the constraints

→ Now, we estimate the probability that a particular clause c_i is satisfied.

Claim: A clause c_i with k literals is satisfied with prob at least $\left(1 - \left(1 - \frac{1}{k}\right)^k\right) v_i$

Assume WLOG all variables in C_i appear in their pure form.

So, $C_i = x_1 \vee x_2 \vee x_3 \ldots x_k$

Teacher's Signature

→ So, in relaxed LP,

$$u_1 + u_2 + \cdots + u_k \geq v_i$$

So our best possible soln satisfies this constraint

So, Now,

C_i is false if all x_j are false &

x_j = false = prob $(1 - u_j)$

⇒ all x_j are false = prob $\prod_j (1 - u_j)$

So, C_i is satisfiable = prob. $1 - \pi(1 - u_j)$

To maximise this prob—

we want to minimise $\pi(1 - u_j)$

given that $\sum u_j \geq v_i$

This can be done when $u_j = \frac{v_i}{k}$

for each j (Proof by calculus)

⇒ Prob for C_i to be satisfiable

$$1 - \left(1 - \frac{v_i}{k}\right)^k$$

Teacher's Signature

→ Now, we want to get rid of v_i because we do not know its value.

To do so,
we prove that a function
$f(x) = 1 - \left(1 - \frac{x}{k}\right)^k$ is concave.
for x in $[0,1]$.
and we know that
if there is a Line L $(0, f(0)), (1, f(1))$
the $f(x) \geq L$ at x

Line value is used as lower bound on func value

$\Rightarrow$ $f(x)$ is at least $1 - \left(1 - \frac{1}{k}\right)^k$
for x in $[0,1]$.

for $x=0$ $(0, f(0)=0)$
for $x=1$ $(1, f(1) = 1-\left(1-\frac{1}{k}\right)^k)$

so eqn of line:
$\rightarrow y = mx+c$
$\rightarrow y = \frac{y_1-y_2}{x_1-x_2}x + c$
$\rightarrow y = \left(1-\left(1-\frac{1}{k}\right)^k\right)x$

$\Rightarrow$ Prob(C_i being satisfiable) $\geq \left(1-\left(1-\frac{1}{k}\right)^k\right) v_i$

→ So, for finding no. of satisfiable clauses we use linearity of expectation

$\geq \sum_{i=1}^{m} (\quad) v_i$

$\geq (\quad) \sum_{i=1}^{m} v_i$

Teacher's Signature

and $\sum_{i=1}^{m} v_i \geq m^*(I)$

So,

$$\sum_i \left(1-\left(1-\frac{1}{k}\right)^k\right) \cdot v_i \geq \left(1-\left(1-\frac{1}{k}\right)^k\right) \times m^*I$$

$$m^A(I) \geq \left(\quad\right) m^*I$$

So, we satisfy at least $1-\left(1-\frac{1}{k}\right)^k$ fraction of maximum no. of clauses that can be satisfied

Lecture 10: 9th February

If we look at diff values of k for the two algo. & the corresponding performance ratio

k	algo 1 = $1-2^{-k}$	algo 2 =
1	0.5	1.0
2	0.75	0.75
3	0.875	0.703
4	0.938	0.684
5	0.969	0.672

Teacher's Signature

we see that
$k \uparrow$ → algo 1 better
$k \downarrow$ → algo 2 better

algo 1 → VAR ↗
algo 2 → LP rounding ↘

→ So, we want to combine the two algo to get a 3/4 ratio

→ when $k=1$; algo 2 has 1 as performance ratio. This means the algo is able to compete ~~c~~ best possible soln — i.e. maxi # clauses satis = clauses satisfied by algo
As when every clause has only one variable you'll be able to satisfy all clauses that have non ~~opes~~ overlapping variable & the ones ~~c~~ overlapping variables if one clause has variable is pure form & another clause has variable is complementary form anyway
best possible soln is satisfying one of these clauses.

So, two ways to combine the 2 —

① run both, pick best soln
② toss a fair coin, pick algo based on outcome

In ① the best is always at least 0.75

$$\geq \frac{3}{4}$$

as we can see from the table.

Teacher's Signature

SHREE
DATE: / /
PAGE NO.:

→ So, we will show that expected performance ratio will be at least 3/4.

let

n_1 → expected # clause satisfied by algo 1 (UAR)

n_2 → ——————— algo 2

So for way 2 –

expected # satisfiable clauses = $\frac{n_1 + n_2}{2}$

$$\frac{1}{2} \times n_1 + \frac{1}{2} \times n_2 = \frac{n_1 + n_2}{2}$$

→ Now we will show that

$$\frac{n_1 + n_2}{2} \geq \frac{3}{4} \sum_i v_i \geq \frac{3}{4} m^*(I)$$

∵ n_1 doesn't relate to v_i; we try to compare n_1 & $\sum_i v_i$

For doing so,
group clauses into buckets depending on the number of variables

$k=1$	$k=2$	...	$k=r$
c_2	c_6 c_4	...	c_1 c_5

where r is max no. of variables in a clause

Teacher's Signature

Now, algo 1 perfoems well for highee k
and
let C_i have k variable
$P(C_i = T) = 1 - 2^{-k}$

So,

n_1 = expected # satisfiable clauses by algo 1

$$= \sum_k \sum_{C_i \text{ has } k \text{ literals}} (1 - 2^{-k})$$

$$\geq \sum_k \sum_{C_i \text{ has } k \text{ literals}} (1 - 2^{-k}) v_j$$

$$(\because 0 \leq v_j \leq 1)$$

and

$$n_2 \geq \sum_k \sum_{C_i \text{ has } k \text{ literals}} \beta_k \cdot v_i$$

where $\beta_k = 1 - \left(1 - \frac{1}{k}\right)^k$

Combing the two —

$$\frac{n_1 + n_2}{2} = \frac{1}{2}\left((1 - 2^{-k}) + \beta_k\right) \cdot v_i$$

Now show that
$(1 - 2^{-k}) + \beta_k$ is at least $\frac{3}{2}$
for $k \geq 1$

Teacher's Signature

We want to show =

$$\Rightarrow \quad 1 - \frac{1}{2^k} + B_k \geq \frac{3}{2}$$

$$\Rightarrow \quad B_k \geq \frac{1}{2} + \frac{1}{2^k}$$

Hence, we need to prove =

$$1 - \left(1 - \frac{1}{k}\right)^k \geq \frac{1}{2} + \frac{1}{2^k}$$

and that's all

Other techniques for MAXSAT is semi-definite programming = 0.78 performance ratio

Hashing

hash table = $|T|$
hash function - $h(x)$

S is subset of universe U

$|S| > |T|$

$|U| >> |T|$

so map can never be one to one.

When x, y in U have $h(x) = h(y)$
↳ collision

Teacher's Signature