Advanced Algorithmic Techniques (COMP523)

Randomised Algorithms 4

Recap and plan

- **Last lecture:**
  - Types of randomised algorithms
  - Randomised approximation algorithms.
    - Applications: MAX-SAT, MAX-3SAT, MAX-CUT
- **This lecture:**
  - Derandomisation using conditional expectations.
  - Randomised Rounding.
    - Application: MAX-SAT

A 2-approximation algorithm for MAX-SAT

- Algorithm: For each variable x_i, set x_i to *1* with probability 1/2 and to *0* with probability 1/2.

Derandomisation

- Sometimes it is possible to "derandomise" a randomised algorithm A_{rand} and obtain a deterministic algorithm A_{det}.
- The performance of A_{det} is the same as the expected performance of A_{rand}.
- We can use randomisation at no extra cost! (except a polynomial running time overhead).
- Different methods for derandomisation.
  - Can be very complicated (*pseudo-random generators*).
  - Can be relatively simple (*conditional expectations*).

Derandomisation

- Algorithm: For each variable x_i, set x_i to *1* with probability 1/2 and to *0* with probability 1/2.

Derandomisation

- Algorithm: For each variable x_i, set x_i to *1* with probability 1/2 and to *0* with probability 1/2.

- Algorithm: Set variable x_i to 1 or 0 *deterministically*, and the remaining variables to *1* with probability 1/2 and to *0* with probability 1/2, as before.

Derandomisation

- Algorithm: For each variable x_i, set x_i to *1* with probability 1/2 and to *0* with probability 1/2.
- Algorithm: Set variable x_i to 1 or 0 *deterministically*, and the remaining variables to *1* with probability 1/2 and to *0* with probability 1/2, as before.
    - How to set x_i?

Derandomisation

- Algorithm: For each variable x_i, set x_i to *1* with probability 1/2 and to *0* with probability 1/2.
- Algorithm: Set variable x_i to 1 or 0 *deterministically*, and the remaining variables to *1* with probability 1/2 and to *0* with probability 1/2, as before.
  - How to set x_i?
  - To maximise the *expected value* W of the algorithm.

Derandomisation

- We have:

$$\mathbb{E}[W] = \mathbb{E}[W|x_1 \leftarrow 1] \cdot \mathbf{Pr}[x_1 \leftarrow 1] + \mathbb{E}[W|x_1 \leftarrow 0] \cdot \mathbf{Pr}[x_1 \leftarrow 0]$$
$$= \frac{1}{2}\left(\mathbb{E}[W|x_1 \leftarrow 1] \cdot + \mathbb{E}[W|x_1 \leftarrow 0]\right)$$

- We set x_1 to *1* if $\mathbb{E}[W|x_1 \leftarrow 1] \geq \mathbb{E}[W|x_1 \leftarrow 1]$ and to *0* otherwise.

- Generally, if b_1 is picked to maximise the conditional expectation, it holds that:

$$\mathbb{E}[W|x_1 \leftarrow b_1] \geq \mathbb{E}[W]$$

Applying this to all variables

- Assume that we have set variables x_1, …, x_i to b_1, … b_i this way.
- We set x_{i+1} to *1* if this holds and to *0* otherwise.

$$\mathbb{E}[W \mid x_1 \leftarrow b_1, x_2 \leftarrow b_2, \ \ldots, x_i \leftarrow b_i, x_{i+1} \leftarrow 1] \geq$$
$$\mathbb{E}[W \mid x_1 \leftarrow b_1, x_2 \leftarrow b_2, \ \ldots, x_i \leftarrow b_i, x_{i+1} \leftarrow 0]$$

- Again, if b_{i+1} is picked to maximise the conditional expectation, it holds that:

$$\mathbb{E}[W \mid x_1 \leftarrow b_1, x_2 \leftarrow b_2, \ \ldots, x_i \leftarrow b_i, x_{i+1} \leftarrow b_{i+1}] \geq \mathbb{E}[W]$$

In the end

In the end

- In the end we have set all variables *deterministically*.

In the end

- In the end we have set all variables *deterministically*.
- We have $\mathbb{E}[W \mid x_1 \leftarrow b_1, x_2 \leftarrow b_2, \ \ldots, \ x_{n-1} \leftarrow b_{n-1}, x_n \leftarrow b_n] \geq \mathbb{E}[W]$

In the end

- In the end we have set all variables *deterministically*.
- We have $\mathbb{E}[W \mid x_1 \leftarrow b_1, x_2 \leftarrow b_2, \ \ldots, \ x_{n-1} \leftarrow b_{n-1}, x_n \leftarrow b_n] \geq \mathbb{E}[W]$
- We know that $\mathbb{E}[W] \geq \frac{1}{2} \cdot OPT$

In the end

- In the end we have set all variables *deterministically*.
- We have $\mathbb{E}[W \mid x_1 \leftarrow b_1, x_2 \leftarrow b_2, \ \ldots, \ x_{n-1} \leftarrow b_{n-1}, x_n \leftarrow b_n] \geq \mathbb{E}[W]$
- We know that $\mathbb{E}[W] \geq \frac{1}{2} \cdot OPT$
- We have devised a *deterministic* 2-approximation algorithm.

In the end

- In the end we have set all variables *deterministically*.
- We have $\mathbb{E}[W \mid x_1 \leftarrow b_1, x_2 \leftarrow b_2, \ \ldots, x_{n-1} \leftarrow b_{n-1}, x_n \leftarrow b_n] \geq \mathbb{E}[W]$
- We know that $\mathbb{E}[W] \geq \frac{1}{2} \cdot OPT$
- We have devised a *deterministic* 2-approximation algorithm.
- Is it polynomial-time?

Computing the expectations

- We have to be able to compute the conditional expectations in polynomial time.

$$\mathbb{E}[W \mid x_1 \leftarrow b_1, \ \ldots, x_i \leftarrow b_i] = \sum_{j=1}^{m} w_j \cdot \mathbb{E}[Y_j \mid x_1 \leftarrow b_1, \ \ldots, x_i \leftarrow b_i]$$

$$= \sum_{j=1}^{m} w_j \cdot \textbf{Pr}[\textbf{clause } C_j \textbf{ is satisfied} \mid x_1 \leftarrow b_1, \ \ldots, x_i \leftarrow b_i]$$

- The probability is
  - *1* if the variables already set satisfy the clause.
  - *1-(1/2)^k* otherwise, where *k* is the set of unset variables.

Method of conditional expectations

- Derandomisation using *conditional expectations*.
- Works for a wide variety of applications as long as
  - The variables are set independently.
  - The conditional expectations can be calculated in polynomial time.

Recall: Deterministic Rounding

- We can solve the LP-relaxation in polynomial time, to find an optimal solution.
- The optimal solution is a “fractional” vertex cover, where variables can take values between *0* and *1*.
- We round the fractional solution to an integer solution.

Recall: Deterministic Rounding

- We can solve the LP-relaxation in polynomial time, to find an optimal solution.
- The optimal solution is a "fractional" vertex cover, where variables can take values between *0* and *1*.
- We round the fractional solution to an integer solution.
  - We pick a variable x_i and we set it to *1* or *0*.

Recall: Deterministic Rounding

- We can solve the LP-relaxation in polynomial time, to find an optimal solution.
- The optimal solution is a “fractional” vertex cover, where variables can take values between *0* and *1*.
- We round the fractional solution to an integer solution.
  - We pick a variable x_i and we set it to *1* or *0*.
  - If we set everything to *0*, it is not a vertex cover.

Recall: Deterministic Rounding

- We can solve the LP-relaxation in polynomial time, to find an optimal solution.
- The optimal solution is a "fractional" vertex cover, where variables can take values between *0* and *1*.
- We round the fractional solution to an integer solution.
  - We pick a variable x_i and we set it to *1* or *0*.
  - If we set everything to *0*, it is not a vertex cover.
  - If we set everything to *1*, we "pay" too much.

Recall: Deterministic Rounding

- We can solve the LP-relaxation in polynomial time, to find an optimal solution.
- The optimal solution is a "fractional" vertex cover, where variables can take values between *0* and *1*.
- We round the fractional solution to an integer solution.
  - We pick a variable x_i and we set it to *1* or *0*.
  - If we set everything to *0*, it is not a vertex cover.
  - If we set everything to *1*, we "pay" too much.
  - We set variable x_i to *1* if $x_i \geq 1/2$ and to *0* otherwise.

Randomised Rounding

- We formulate the problem as an ILP.
- We write the LP-relaxation.
- We solve the LP-relaxation.
- We *round* the variables with probabilities that can depend on their values.

MAX SAT as an ILP

MAX SAT as an ILP

Variables: y_i = *1* if x_i is *true* and *0* otherwise.

MAX SAT as an ILP

Variables: y_i = *1* if x_i is *true* and *0* otherwise.

We denote clause C_j by $$\bigvee_{i \in P_j} x_i \vee \bigvee_{i \in N_j} \bar{x}_i$$

MAX SAT as an ILP

Variables: y_i = *1* if x_i is *true* and *0* otherwise.

We denote clause C_j by $\bigvee_{i \in P_j} x_i \vee \bigvee_{i \in N_j} \bar{x}_i$

Variables: z_j = *1* if clause C_j is *satisfied* and *0* otherwise.

MAX SAT as an ILP

Variables: y_i = *1* if x_i is *true* and *0* otherwise.

We denote clause C_j by $$\bigvee_{i \in P_j} x_i \vee \bigvee_{i \in N_j} \bar{x}_i$$

Variables: z_j = *1* if clause C_j is *satisfied* and *0* otherwise.

We have the inequality: $$\sum_{i \in P_j} y_i + \sum_{i \in N_j} (1 - y_i) \geq z_j$$

MAX SAT as an ILP

maximise $$\sum_{j=1}^{m} w_j \cdot z_j$$

subject to $$\sum_{i \in P_j} y_i + \sum_{i \in N_j} (1 - y_i) \geq z_j$$ **for all** $C_j = \bigvee_{i \in P_j} x_i \vee \bigvee_{i \in N_j} \bar{x}_i$

$$y_i \in \{0,1\} \qquad i = 1,\ldots,n$$

$$0 \leq z_j \leq 1 \qquad j = 1,\ldots,m$$

MAX SAT LP-relaxation

maximise $$\sum_{j=1}^{m} w_j \cdot z_j$$

subject to $$\sum_{i \in P_j} y_i + \sum_{i \in N_j} (1 - y_i) \geq z_j \qquad \textbf{for all } C_j = \bigvee_{i \in P_j} x_i \vee \bigvee_{i \in N_j} \bar{x}_i$$

$$0 \leq y_i \leq 1 \qquad i = 1, \ldots, n$$

$$0 \leq z_j \leq 1 \qquad j = 1, \ldots, m$$

Randomised Rounding

- Let (y^*, z^*) be a solution to the LP-relaxation.

Randomised Rounding

- Let (y^*, z^*) be a solution to the LP-relaxation.
- Rounding: Set x_i to *true* independently with probability y_i^*.

Randomised Rounding

- Let (y^*, z^*) be a solution to the LP-relaxation.
- Rounding: Set x_i to *true* independently with probability y_i^*.
  - e.g., if $y^* = (1/3, 1/4, 5/6, 1/2, \ldots)$ we will set variables $x_1, x_2, x_3, x_4, \ldots$ to *true* with probabilities *1/3*, *1/4*, *5/6*, *1/2*, … respectively.

Analysis

Analysis

$$\textbf{Pr}[\textbf{clause } C_j \textbf{ is not satisfied}] = \prod_{i \in P_j} (1 - y_i^*) + \prod_{i \in N_j} y_i^*$$

Analysis

$$\textbf{Pr}[\textbf{clause } C_j \textbf{ is not satisfied}] = \prod_{i \in P_j} (1 - y_i^*) + \prod_{i \in N_j} y_i^*$$

$$\leq \left[\frac{1}{\ell_j} \left(\sum_{i \in P_j} (1 - y_i^*) + \sum_{i \in N_j} y_i^* \right) \right]^{\ell_j}$$

Analysis

$$\textbf{Pr}[\textbf{clause } C_j \textbf{ is not satisfied}] = \prod_{i \in P_j} (1 - y_i^*) + \prod_{i \in N_j} y_i^*$$

$$\leq \left[\frac{1}{\ell_j} \left(\sum_{i \in P_j} (1 - y_i^*) + \sum_{i \in N_j} y_i^* \right) \right]^{\ell_j}$$

Number of literals in clause C_j.

Analysis

Arithmetic-Geometric Mean Inequality

$$\left(\prod_{i=1}^{k} a_i\right)^k \le \frac{1}{k}\sum_{i=1}^{k} a_i$$

$$\textbf{Pr[clause } C_j \textbf{ is not satisfied]} = \prod_{i \in P_j}(1 - y_i^*) + \prod_{i \in N_j} y_i^*$$

$$\le \left[\frac{1}{\ell_j}\left(\sum_{i \in P_j}(1 - y_i^*) + \sum_{i \in N_j} y_i^*\right)\right]^{\ell_j}$$

Number of literals in clause C_j.

Analysis

Arithmetic-Geometric Mean Inequality

$$\left(\prod_{i=1}^{k} a_i\right)^{k} \leq \frac{1}{k}\sum_{i=1}^{k} a_i$$

$$\textbf{Pr}[\textbf{clause } C_j \textbf{ is not satisfied}] = \prod_{i \in P_j}(1 - y_i^*) + \prod_{i \in N_j} y_i^*$$

$$\leq \left[\frac{1}{\ell_j}\left(\sum_{i \in P_j}(1 - y_i^*) + \sum_{i \in N_j} y_i^*\right)\right]^{\ell_j}$$

Number of literals in clause C_j.

$$\leq \left[1 - \frac{1}{\ell_j}\left(\sum_{i \in P_j} y_i^* + \sum_{i \in N_j}(1 - y_i^*)\right)\right]^{\ell_j}$$

Analysis

Arithmetic-Geometric Mean Inequality

$$\left(\prod_{i=1}^{k} a_i\right)^{k} \leq \frac{1}{k}\sum_{i=1}^{k} a_i$$

$$\textbf{Pr}[\textbf{clause } C_j \textbf{ is not satisfied}] = \prod_{i\in P_j}(1-y_i^*) + \prod_{i\in N_j} y_i^*$$

$$\leq \left[\frac{1}{\ell_j}\left(\sum_{i\in P_j}(1-y_i^*) + \sum_{i\in N_j} y_i^*\right)\right]^{\ell_j}$$

Number of literals in clause C_j.

$$\leq \left[1 - \frac{1}{\ell_j}\left(\sum_{i\in P_j} y_i^* + \sum_{i\in N_j}(1-y_i^*)\right)\right]^{\ell_j}$$

$$\leq \left(1 - \frac{z_j^*}{\ell_j}\right)^{\ell_j}$$

Analysis

Arithmetic-Geometric Mean Inequality

$$\left(\prod_{i=1}^{k} a_i\right)^{k} \leq \frac{1}{k}\sum_{i=1}^{k} a_i$$

Pr[**clause** C_j **is not satisfied**] $= \prod_{i \in P_j}(1 - y_i^*) + \prod_{i \in N_j} y_i^*$

$$\leq \left[\frac{1}{\ell_j}\left(\sum_{i \in P_j}(1 - y_i^*) + \sum_{i \in N_j} y_i^*\right)\right]^{\ell_j}$$

Number of literals in clause C_j.

$$\leq \left[1 - \frac{1}{\ell_j}\left(\sum_{i \in P_j} y_i^* + \sum_{i \in N_j}(1 - y_i^*)\right)\right]^{\ell_j}$$

$$\leq \left(1 - \frac{z_j^*}{\ell_j}\right)^{\ell_j}$$

why?

MAX SAT as an ILP

Variables: y_i = *1* if x_i is *true* and *0* otherwise.

We denote clause C_j by $$\bigvee_{i \in P_j} x_i \vee \bigvee_{i \in N_j} \bar{x}_i$$

Variables: z_j = *1* if clause C_j is *satisfied* and *0* otherwise.

We have the inequality: $$\sum_{i \in P_j} y_i + \sum_{i \in N_j} (1 - y_i) \geq z_j$$

MAX SAT as an ILP

Variables: y_i = *1* if x_i is *true* and *0* otherwise.

We denote clause C_j by $$\bigvee_{i \in P_j} x_i \vee \bigvee_{i \in N_j} \bar{x}_i$$

Variables: z_j = *1* if clause C_j is *satisfied* and *0* otherwise.

We have the inequality: $$\sum_{i \in P_j} y_i + \sum_{i \in N_j} (1 - y_i) \geq z_j$$

MAX SAT as an ILP

Variables: y_i = *1* if x_i is *true* and *0* otherwise.

We denote clause C_j by $\bigvee_{i \in P_j} x_i \vee \bigvee_{i \in N_j} \bar{x}_i$

Variables: z_j = *1* if clause C_j is *satisfied* and *0* otherwise.

We have the inequality:

$$\sum_{i \in P_j} y_i + \sum_{i \in N_j} (1 - y_i) \geq z_j$$

$$\sum_{i \in P_j} y_i^* + \sum_{i \in N_j} (1 - y_i^*) \geq z_j^*$$

Analysis

Arithmetic-Geometric Mean Inequality

$$\left(\prod_{i=1}^{k} a_i\right)^k \leq \frac{1}{k}\sum_{i=1}^{k} a_i$$

$$\textbf{Pr}[\textbf{clause } C_j \textbf{ is not satisfied}] = \prod_{i \in P_j}(1 - y_i^*)\prod_{i \in N_j} y_i^*$$

$$\leq \left[\frac{1}{\ell_j}\left(\sum_{i \in P_j}(1 - y_i^*) + \sum_{i \in N_j} y_i^*\right)\right]^{\ell_j}$$

Number of literals in clause C_j.

$$\leq \left[1 - \frac{1}{\ell_j}\left(\sum_{i \in P_j} y_i^* + \sum_{i \in N_j}(1 - y_i^*)\right)\right]^{\ell_j}$$

$$\leq \left(1 - \frac{z_j^*}{\ell_j}\right)^{\ell_j}$$

Analysis

$$\textbf{Pr}[\textbf{clause } C_j \textbf{ is not satisfied}] = \prod_{i \in P_j} (1 - y_i^*) \prod_{i \in N_j} y_i^*$$

$$\leq \left(1 - \frac{z_j^*}{\ell_j}\right)^{\ell_j}$$

Analysis

$$\textbf{Pr}[\textbf{clause } C_j \textbf{ is not satisfied}] = \prod_{i \in P_j} (1 - y_i^*) \prod_{i \in N_j} y_i^*$$

$$\leq \left(1 - \frac{z_j^*}{\ell_j}\right)^{\ell_j}$$

$$\textbf{Pr}[\textbf{clause } C_j \textbf{ is satisfied}] \geq 1 - \left(1 - \frac{z_j^*}{\ell_j}\right)^{\ell_j}$$

Analysis

Pr[**clause** C_j **is not satisfied**] $= \prod_{i \in P_j} (1 - y_i^*) \prod_{i \in N_j} y_i^*$

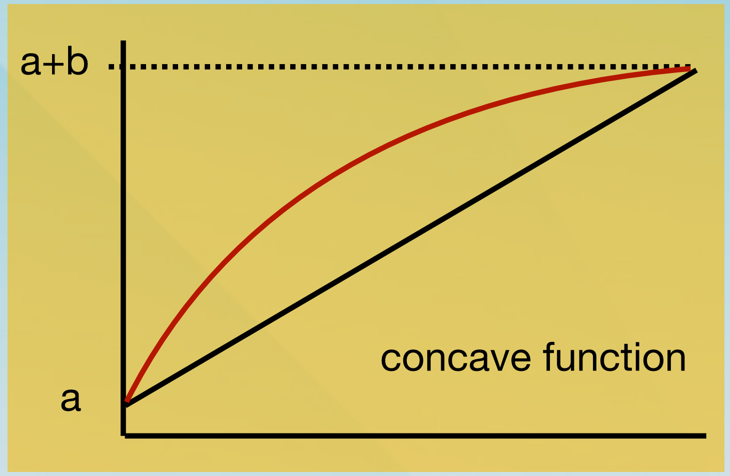
$$\leq \left(1 - \frac{z_j^*}{\ell_j}\right)^{\ell_j}$$

Pr[**clause** C_j **is satisfied**]

$$\geq 1 - \left(1 - \frac{z_j^*}{\ell_j}\right)^{\ell_j}$$

$$\geq \left[1 - \left(1 - \frac{1}{\ell_j}\right)^{\ell_j}\right] z_j^*$$

Analysis



$$\textbf{Pr}[\textbf{clause } C_j \textbf{ is not satisfied}] = \prod_{i \in P_j} (1 - y_i^*) \prod_{i \in N_j} y_i^*$$

$$\leq \left(1 - \frac{z_j^*}{\ell_j}\right)^{\ell_j}$$

$$\textbf{Pr}[\textbf{clause } C_j \textbf{ is satisfied}] \geq 1 - \left(1 - \frac{z_j^*}{\ell_j}\right)^{\ell_j}$$

$$\geq \left[1 - \left(1 - \frac{1}{\ell_j}\right)^{\ell_j}\right] z_j^*$$

Analysis

$$\mathbb{E}[W] = \sum_{j=1}^{m} w_j \cdot \textbf{Pr}[\textbf{clause } C_j \textbf{ is satisfied}]$$

Analysis

$$\mathbb{E}[W] = \sum_{j=1}^{m} w_j \cdot \textbf{Pr}[\textbf{clause } C_j \textbf{ is satisfied}]$$

$$\geq \sum_{j=1}^{m} w_j \cdot z_j^* \cdot \left[1 - \left(1 - \frac{1}{\ell_j}\right)^{\ell_j}\right]$$

Analysis

$$\mathbb{E}[W] = \sum_{j=1}^{m} w_j \cdot \textbf{Pr}[\textbf{clause } C_j \textbf{ is satisfied}]$$

$$\geq \sum_{j=1}^{m} w_j \cdot z_j^* \cdot \left[1 - \left(1 - \frac{1}{\ell_j}\right)^{\ell_j}\right]$$

$$\geq \min_{k \geq 1} \left[1 - \left(1 - \frac{1}{k}\right)^{k}\right] \sum_{j=1}^{m} w_j \cdot z_j^*$$

Analysis

$$\mathbb{E}[W] = \sum_{j=1}^{m} w_j \cdot \textbf{Pr}[\textbf{clause } C_j \textbf{ is satisfied}]$$

$$\geq \sum_{j=1}^{m} w_j \cdot z_j^* \cdot \left[1 - \left(1 - \frac{1}{\ell_j}\right)^{\ell_j}\right]$$

$$\geq \min_{k \geq 1} \left[1 - \left(1 - \frac{1}{k}\right)^{k}\right] \sum_{j=1}^{m} w_j \cdot z_j^*$$

$$\geq \left(1 - \frac{1}{e}\right) \sum_{j=1}^{m} w_j \cdot z_j^*$$

Analysis

$$\mathbb{E}[W] = \sum_{j=1}^{m} w_j \cdot \textbf{Pr}[\textbf{clause } C_j \textbf{ is satisfied}]$$

$$\geq \sum_{j=1}^{m} w_j \cdot z_j^* \cdot \left[1 - \left(1 - \frac{1}{\ell_j}\right)^{\ell_j}\right]$$

$$\geq \min_{k \geq 1} \left[1 - \left(1 - \frac{1}{k}\right)^{k}\right] \sum_{j=1}^{m} w_j \cdot z_j^*$$

$$\geq \left(1 - \frac{1}{e}\right) \sum_{j=1}^{m} w_j \cdot z_j^*$$

$$\geq \left(1 - \frac{1}{e}\right) OPT$$

Randomised Rounding for MAX-SAT

- Our randomised algorithm gives an approximation ratio of $1/(1-1/e) \approx 1.59$.

Randomised Rounding for MAX-SAT

- Our randomised algorithm gives an approximation ratio of $1/(1-1/e) \approx 1.59$.
- This is better than 2.

Randomised Rounding for MAX-SAT

- Our randomised algorithm gives an approximation ratio of $1/(1-1/e) \approx 1.59$.
- This is better than 2.
- This is better than 1.618. (why this?)

Randomised Rounding for MAX-SAT

- Our randomised algorithm gives an approximation ratio of $1/(1-1/e) \approx 1.59$.
- This is better than 2.
- This is better than 1.618. (why this?)
    - Sidenote: $1.618 = \phi$.

The better of the two

Algorithm 1: $\textbf{Pr}[\textbf{clause } C_j \textbf{ is satisfied} \geq \left(1 - \left(\frac{1}{2}\right)^{\ell_j}\right)$

Algorithm 2: $\textbf{Pr}[\textbf{clause } C_j \textbf{ is satisfied}] \geq \left[1 - \left(1 - \frac{1}{\ell_j}\right)^{\ell_j}\right] z_j^*$

The better of the two

Algorithm 1: **Pr**[**clause** C_j **is satisfied** $\geq \left(1 - \left(\frac{1}{2}\right)^{\ell_j}\right)$

Algorithm 2: **Pr**[**clause** C_j **is satisfied**] $\geq \left[1 - \left(1 - \frac{1}{\ell_j}\right)^{\ell_j}\right] z_j^*$

If the clause is *short*, Algorithm 1 performs well.
If the clause is *long*, Algorithm 2 performs well.

The better of the two

Algorithm 1: **Pr[clause C_j is satisfied** $\geq \left(1 - \left(\frac{1}{2}\right)^{\ell_j}\right)$

Algorithm 2: **Pr[clause C_j is satisfied]** $\geq \left[1 - \left(1 - \frac{1}{\ell_j}\right)^{\ell_j}\right] z_j^*$

If the clause is *short*, Algorithm 1 performs well.
If the clause is *long*, Algorithm 2 performs well.

Algorithm 3: Choose the *better* of Algorithm 1 and Algorithm 2.

Analysis

Analysis

$\mathbb{E}[W] = \mathbb{E}[\max(W_1, W_2)]$

Analysis

$$\mathbb{E}[W] = \mathbb{E}[\max(W_1, W_2)]$$

$$\geq \mathbb{E}\left[\frac{1}{2}W_1 + \frac{1}{2}W_2\right]$$

Analysis

$$\mathbb{E}[W] = \mathbb{E}[\max(W_1, W_2)]$$

$$\geq \mathbb{E}\left[\frac{1}{2}W_1 + \frac{1}{2}W_2\right]$$

$$\geq \frac{1}{2}\mathbb{E}[W_1] + \frac{1}{2}\mathbb{E}[W_2]$$

Analysis

$$
\begin{aligned}
\mathbb{E}[W] &= \mathbb{E}[\max(W_1, W_2)] \\
&\geq \mathbb{E}\left[\frac{1}{2}W_1 + \frac{1}{2}W_2\right] \\
&\geq \frac{1}{2}\mathbb{E}[W_1] + \frac{1}{2}\mathbb{E}[W_2] \\
&\geq \frac{1}{2}\sum_{j=1}^{m} w_j \left[1 - \left(\frac{1}{2}\right)^{\ell_j}\right] + \frac{1}{2}\sum_{j=1}^{m} w_j \cdot z_j^* \left[1 - \left(1 - \frac{1}{\ell_j}\right)^{\ell_j}\right]
\end{aligned}
$$

Analysis

$$
\begin{aligned}
\mathbb{E}[W] = \mathbb{E}[\max(W_1, W_2)] \\
&\geq \mathbb{E}\left[\frac{1}{2}W_1 + \frac{1}{2}W_2\right] \\
&\geq \frac{1}{2}\mathbb{E}[W_1] + \frac{1}{2}\mathbb{E}[W_2] \\
&\geq \frac{1}{2}\sum_{j=1}^{m} w_j\left[1 - \left(\frac{1}{2}\right)^{\ell_j}\right] + \frac{1}{2}\sum_{j=1}^{m} w_j \cdot z_j^* \left[1 - \left(1 - \frac{1}{\ell_j}\right)^{\ell_j}\right] \\
&\geq \sum_{j=1}^{m} w_j \cdot z_j^* \left[\frac{1}{2}\left(1 - \left(\frac{1}{2}\right)^{\ell_j}\right) + \frac{1}{2}\left(1 - \left(1 - \frac{1}{\ell_j}\right)^{\ell_j}\right)\right]
\end{aligned}
$$

Analysis

This quantity: $$\frac{1}{2}\left(1-\left(\frac{1}{2}\right)^{\ell_j}\right)+\frac{1}{2}\left(1-\left(1-\frac{1}{\ell_j}\right)^{\ell_j}\right)$$

Analysis

This quantity: $$\frac{1}{2}\left(1-\left(\frac{1}{2}\right)^{\ell_j}\right)+\frac{1}{2}\left(1-\left(1-\frac{1}{\ell_j}\right)^{\ell_j}\right)$$

For $\ell_j = 1$, it evaluates to 3/4.

Analysis

This quantity: $$\frac{1}{2}\left(1-\left(\frac{1}{2}\right)^{\ell_j}\right)+\frac{1}{2}\left(1-\left(1-\frac{1}{\ell_j}\right)^{\ell_j}\right)$$

For $\ell_j = 1$, it evaluates to 3/4.

For $\ell_j = 2$, it evaluates to 3/4.

Analysis

This quantity: $$\frac{1}{2}\left(1-\left(\frac{1}{2}\right)^{\ell_j}\right)+\frac{1}{2}\left(1-\left(1-\frac{1}{\ell_j}\right)^{\ell_j}\right)$$

For $\ell_j = 1$, it evaluates to 3/4.

For $\ell_j = 2$, it evaluates to 3/4.

For $\ell_j \geq 3$, we have: $$\left(1-\left(\frac{1}{2}\right)^{\ell_j}\right) \geq \frac{7}{8} \qquad \left(1-\left(1-\frac{1}{\ell_j}\right)^{\ell_j}\right) \geq 1-\frac{1}{e}$$

Analysis

This quantity: $$\frac{1}{2}\left(1-\left(\frac{1}{2}\right)^{\ell_j}\right)+\frac{1}{2}\left(1-\left(1-\frac{1}{\ell_j}\right)^{\ell_j}\right)$$

For $\ell_j = 1$, it evaluates to 3/4.

For $\ell_j = 2$, it evaluates to 3/4.

For $\ell_j \geq 3$, we have: $$\left(1-\left(\frac{1}{2}\right)^{\ell_j}\right) \geq \frac{7}{8} \qquad \left(1-\left(1-\frac{1}{\ell_j}\right)^{\ell_j}\right) \geq 1-\frac{1}{e}$$

And: $$\frac{1}{2}\left(1-\frac{1}{e}\right)+\frac{1}{2}\cdot 78 \approx 0.753 \geq \frac{3}{4}$$

Algorithms for MAX-SAT

- Our randomised algorithm gives an approximation ratio of *$1/(1-1/e)$* ≈ 1.59.
- This is better than 2.
- This is better than 1.618. (why this?)
    - Sidenote: 1.618 = ϕ.

Algorithms for MAX-SAT

- Our randomised algorithm gives an approximation ratio of $1/(1-1/e) \approx 1.59$.
- This is better than 2.
- This is better than 1.618. (why this?)
    - Sidenote: $1.618 = \phi$.
- The "*better of the two*" algorithm has approximation ratio $4/3 \approx 1.33$.

Back to Randomised Rounding

- Is our RR algorithm the best possible?

Back to Randomised Rounding

- Is our RR algorithm the best possible?
- How do we (attempt to) show that?

Back to Randomised Rounding

- Is our RR algorithm the best possible?
- How do we (attempt to) show that?
    - Integrality gap.

Integrality Gap of MAX-SAT

Integrality Gap of MAX-SAT

- Consider the formula:

  $(x_1 \lor x_2) \land (\neg x_1 \lor x_2) \land (x_1 \lor \neg x_2) \land (\neg x_1 \lor \neg x_2)$

Integrality Gap of MAX-SAT

- Consider the formula:

  $(x_1 \vee x_2) \wedge (\neg x_1 \vee x_2) \wedge (x_1 \vee \neg x_2) \wedge (\neg x_1 \vee \neg x_2)$

- The optimal *integral* solution satisfied 3 clauses.

Integrality Gap of MAX-SAT

- Consider the formula:

  $(x_1 \vee x_2) \wedge (\neg x_1 \vee x_2) \wedge (x_1 \vee \neg x_2) \wedge (\neg x_1 \vee \neg x_2)$

- The optimal *integral* solution satisfied 3 clauses.

- The optimal fractional solution sets

  $y_1 = y_2 = 1/2$ and $z_j = 1$ for all j

  and satisfies 4 clauses.

Integrality Gap of MAX-SAT

- Consider the formula:

  $(x_1 \vee x_2) \wedge (\neg x_1 \vee x_2) \wedge (x_1 \vee \neg x_2) \wedge (\neg x_1 \vee \neg x_2)$

- The optimal *integral* solution satisfied 3 clauses.
- The optimal fractional solution sets

  $y_1 = y_2 = 1/2$ and $z_j = 1$ for all j

  and satisfies 4 clauses.

- The integrality gap is at least *4/3*.

What does this mean?

What does this mean?

- We can not hope to design an LP-relaxation and rounding-based algorithm (for this ILP formulation) that outperforms our "*better of the two*" algorithm.

What does this mean?

- We can not hope to design an LP-relaxation and rounding-based algorithm (for this ILP formulation) that outperforms our "*better of the two*" algorithm.
- Can we design one that matches the 4/3 approximation ratio?

What does this mean?

- We can not hope to design an LP-relaxation and rounding-based algorithm (for this ILP formulation) that outperforms our "*better of the two*" algorithm.
- Can we design one that matches the 4/3 approximation ratio?
- Yes we can!

What does this mean?

- We can not hope to design an LP-relaxation and rounding-based algorithm (for this ILP formulation) that outperforms our "*better of the two*" algorithm.
- Can we design one that matches the 4/3 approximation ratio?
- Yes we can!
  - Instead of "Set x_i to *true* independently with probability y_i^*",

What does this mean?

- We can not hope to design an LP-relaxation and rounding-based algorithm (for this ILP formulation) that outperforms our "*better of the two*" algorithm.
- Can we design one that matches the 4/3 approximation ratio?
- Yes we can!
  - Instead of "Set x_i to *true* independently with probability y_i^*",
  - We use "Set x_i to *true* independently with probability $f(y_i^*)$, for some function f.

What does this mean?

- We can not hope to design an LP-relaxation and rounding-based algorithm (for this ILP formulation) that outperforms our "*better of the two*" algorithm.
- Can we design one that matches the 4/3 approximation ratio?
- Yes we can!
  - Instead of "Set x_i to *true* independently with probability y_i^*",
  - We use "Set x_i to *true* independently with probability $f(y_i^*)$, for some function f.
  - Which function f?

What does this mean?

- We can not hope to design an LP-relaxation and rounding-based algorithm (for this ILP formulation) that outperforms our "*better of the two*" algorithm.
- Can we design one that matches the 4/3 approximation ratio?
- Yes we can!
  - Instead of "Set x_i to *true* independently with probability y_i^*",
  - We use "Set x_i to *true* independently with probability $f(y_i^*)$, for some function f.
  - Which function f?
    - Any function such that $1 - 4^{-x} \leq f(x) \leq 4^{x-1}$

Analysis

Analysis

$$\textbf{Pr}[\textbf{clause } C_j \textbf{ is not satisfied}] = \prod_{i \in P_j} (1 - f(y_i^*)) \prod_{i \in N_j} f(y_i^*)$$

Analysis

$$\textbf{Pr}[\textbf{clause } C_j \textbf{ is not satisfied}] = \prod_{i \in P_j} (1 - f(y_i^*)) \prod_{i \in N_j} f(y_i^*)$$

$$\leq \prod_{i \in P_j} 4^{-y_i^*} \prod_{i \in N_j} 4^{y_i^* - 1}$$

Analysis

$$1 - 4^{-x} \le f(x) \le 4^{x-1}$$

Pr[**clause** C_j **is not satisfied**] $= \prod_{i \in P_j} (1 - f(y_i^*)) \prod_{i \in N_j} f(y_i^*)$

$$\le \prod_{i \in P_j} 4^{-y_i^*} \prod_{i \in N_j} 4^{y_i^* - 1}$$

Analysis

$$1 - 4^{-x} \le f(x) \le 4^{x-1}$$

Pr$[$**clause** C_j **is not satisfied**$] = \prod_{i \in P_j} (1 - f(y_i^*)) \prod_{i \in N_j} f(y_i^*)$

$$\le \prod_{i \in P_j} 4^{-y_i^*} \prod_{i \in N_j} 4^{y_i^* - 1}$$

$$= 4^{-\left(\sum_{i \in P_j} y_i^* + \sum_{i \in N_j} (1 - y_i^*) \right)}$$

Analysis

$$1 - 4^{-x} \leq f(x) \leq 4^{x-1}$$

Pr[**clause** C_j **is not satisfied**] $= \prod_{i \in P_j} (1 - f(y_i^*)) \prod_{i \in N_j} f(y_i^*)$

$$\leq \prod_{i \in P_j} 4^{-y_i^*} \prod_{i \in N_j} 4^{y_i^* - 1}$$

$$= 4^{-\left(\sum_{i \in P_j} y_i^* + \sum_{i \in N_j} (1 - y_i^*) \right)}$$

$$\leq 4^{-z_j^*}$$

Analysis

$$\sum_{i \in P_j} y_i^* + \sum_{i \in N_j} (1 - y_i^*) \geq z_j^*$$

$$1 - 4^{-x} \leq f(x) \leq 4^{x-1}$$

Pr[**clause** C_j **is not satisfied**] $= \prod_{i \in P_j} (1 - f(y_i^*)) \prod_{i \in N_j} f(y_i^*)$

$$\leq \prod_{i \in P_j} 4^{-y_i^*} \prod_{i \in N_j} 4^{y_i^* - 1}$$

$$= 4^{-\left(\sum_{i \in P_j} y_i^* + \sum_{i \in N_j} (1 - y_i^*) \right)}$$

$$\leq 4^{-z_j^*}$$

Analysis

$$\sum_{i \in P_j} y_i^* + \sum_{i \in N_j} (1 - y_i^*) \geq z_j^*$$

$$1 - 4^{-x} \leq f(x) \leq 4^{x-1}$$

Pr[**clause** C_j **is not satisfied**] $= \prod_{i \in P_j} (1 - f(y_i^*)) \prod_{i \in N_j} f(y_i^*)$

$$\leq \prod_{i \in P_j} 4^{-y_i^*} \prod_{i \in N_j} 4^{y_i^* - 1}$$

$$= 4^{-\left(\sum_{i \in P_j} y_i^* + \sum_{i \in N_j} (1 - y_i^*) \right)}$$

$$\leq 4^{-z_j^*}$$

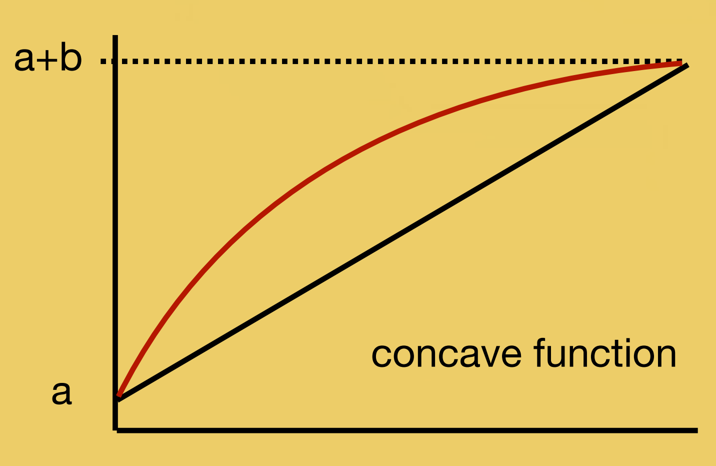
Pr[**clause** C_j **is satisfied**] $\geq 1 - 4^{-z_j^*} \geq \left(1 - \frac{1}{4}\right) z_j^* = \frac{3}{4} z_j^*$

Analysis

$$\sum_{i \in P_j} y_i^* + \sum_{i \in N_j} (1 - y_i^*) \geq z_j^*$$

$$1 - 4^{-x} \leq f(x) \leq 4^{x-1}$$

Pr[clause C_j **is not satisfied]** $= \prod_{i \in P_j} (1 - f(y_i^*)) \prod_{i \in N_j} f(y_i^*)$



$$\leq \prod_{i \in P_j} 4^{-y_i^*} \prod_{i \in N_j} 4^{y_i^* - 1}$$

$$= 4^{-\left(\sum_{i \in P_j} y_i^* + \sum_{i \in N_j} (1 - y_i^*) \right)}$$

$$\leq 4^{-z_j^*}$$

Pr[clause C_j **is satisfied]** $\geq 1 - 4^{-z_j^*} \geq \left(1 - \frac{1}{4}\right) z_j^* = \frac{3}{4} z_j^*$

Analysis

$$\mathbb{E}[W] = \sum_{j=1}^{m} w_j \cdot \textbf{Pr}[\textbf{clause } C_j \textbf{ is satisfied}]$$

Analysis

$$\mathbb{E}[W] = \sum_{j=1}^{m} w_j \cdot \textbf{Pr}[\textbf{clause } C_j \textbf{ is satisfied}]$$

$$\geq \sum_{j=1}^{m} \frac{3}{4} w_j \cdot z_j^*$$

Analysis

$$\mathbb{E}[W] = \sum_{j=1}^{m} w_j \cdot \textbf{Pr}[\textbf{clause } C_j \textbf{ is satisfied}]$$

$$\geq \sum_{j=1}^{m} \frac{3}{4} w_j \cdot z_j^*$$

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

Analysis

$$\mathbb{E}[W] = \sum_{j=1}^{m} w_j \cdot \textbf{Pr}[\textbf{clause } C_j \textbf{ is satisfied}]$$

$$\geq \sum_{j=1}^{m} \frac{3}{4} w_j \cdot z_j^*$$

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

Remark: Other choices of the function f work as well.

Algorithms for MAX-SAT

- Our randomised algorithm gives an approximation ratio of *1/(1-1/e)* ≈ 1.59.
- This is better than 2.
- This is better than 1.618. (why this?)
    - Sidenote: 1.618 = ϕ.
- The "*better of the two*" algorithm has approximation ratio *4/3* ≈ 1.33.

Algorithms for MAX-SAT

- Our randomised algorithm gives an approximation ratio of *1/(1-1/e)* ≈ 1.59.
- This is better than 2.
- This is better than 1.618. (why this?)
  - Sidenote: 1.618 = ϕ.
- The "*better of the two*" algorithm has approximation ratio *4/3* ≈ 1.33.
- The more sophisticated RR algorithm has an approximation ratio of *4/3* ≈ 1.33.