



If we tell an information to B,

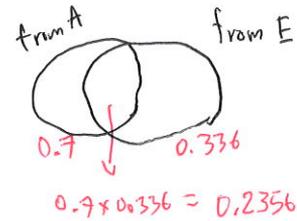
- B will know the information with prob. 1
- A will know the information with prob. 0.9
- C will know the information with prob. 0.8
- D will know the information with prob.  $0.8 \times 0.8 = 0.64$
- E will know the information with prob.  $0.8 \times 0.8 \times 0.4 = 0.256$

Expected # persons  
 = Influence of B  
 =  $1 + 0.9 + 0.8 + 0.64 + 0.256$   
 = 3.60

$\therefore$  C is more influential than B.

If we tell an information to A and E

- A will know the information with prob. 1
- E will know the information with prob. 1
- B will know the information from A with prob. 0.7
- B will know the information from E with prob.  $0.6 \cdot 0.7 \cdot 0.8 = 0.336$
- B will know the information from A or E with prob.  $0.7 + 0.336 - 0.2356 = 0.8$
- C will know the information with prob. 0.7448
- D will know the information with prob. 0.78



Influence of A, E =  $1 + 1 + 0.7 + 0.336 + 0.8 + 0.7448 + 0.78 = 4.33$

Problem

Input: A social network  $(V, E)$

For each link  $e \in E$ ,  $p(e)$  and budget integer  $k$

Output: A set of persons to tell an information to  $S \subseteq V$

Constraint:  $|S| \leq k$

Objective Function: Maximize expected # persons that know the information.  $f(S)$

Bonus question When  $k=2$ , what is the best solution for our network?

Restate the problem

Input: function  $f: 2^V \rightarrow \mathbb{R}$ , budget  $k$   
 $f(S) :=$  expected # persons

Output:  $S \subseteq V$

Constraint:  $|S| \leq k$

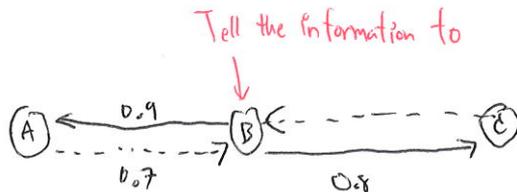
Objective Function: Maximize  $f(S)$

$f$  is a monotone submodular function  $\Rightarrow$  submodular function maximization with size constraint can use greedy algorithm and have

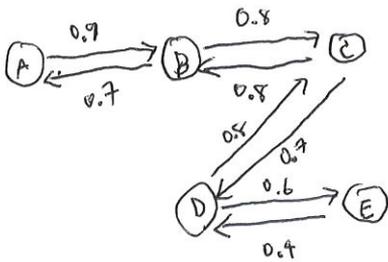
True ~~False~~ because we can reach more persons with a larger set  $S$ .

$$f(SOL) \geq 0.63 \cdot f(OPT)$$

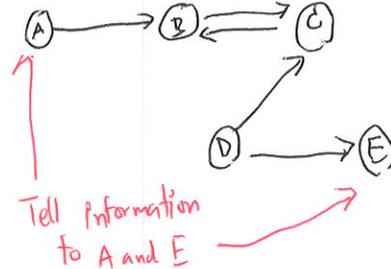
Submodularity



$\rightarrow$  This link is used by B with prob. 0.8  
 The link exists with prob. 0.8



G



- Reach  $A, B, C, E$
- Prob. that we have this network  
 $= 0.9 \cdot 0.3 \cdot 0.8 \cdot 0.8 \cdot 0.3 \cdot 0.6 \cdot 0.6$

$\uparrow$  We have  $A \rightarrow B$   
 $\uparrow$  we don't have  $B \rightarrow A$

$$f(S) = \sum_{G' \text{ possible graphs}} \Pr[G'] \cdot (\# \text{ persons we can reach by } S \text{ in graph } G')$$

$f_{G'}(S)$

Lemma  $f_{G'}$  is a monotone submodular function for all  $G'$

Proof:  $P_i :=$  set of nodes that can reach from  $i$  in graph  $G'$

$$f_{G'}(S) = \left| \bigcup_{i \in S} P_i \right| \leftarrow \text{Proved to be a monotone submodular function on last week.} \quad \square$$

Theorem  $f$  is a monotone submodular function.

Proof Assume that  $S \subseteq S'$ .

$$\begin{aligned} \Delta_e f(S) &= f(S \cup \{e\}) - f(S) \\ &= \sum_{G': \text{possible graphs}} \Pr[G'] \cdot f_{G'}(S \cup \{e\}) - \sum_{G': \text{possible graphs}} \Pr[G'] \cdot f_{G'}(S) \\ &= \sum_{G': \text{possible graphs}} \Pr[G'] (f_{G'}(S \cup \{e\}) - f_{G'}(S)) \\ &= \sum_{G': \text{possible graphs}} \Pr[G'] \cdot \Delta_e f_{G'}(S) \quad \xrightarrow{\text{submodular function}} \\ &\geq \sum_{G': \text{possible graphs}} \Pr[G'] \cdot \Delta_e f_{G'}(S') \quad \geq \Delta_e f_{G'}(S') \\ &= \sum_{G': \text{possible graphs}} \Pr[G'] (f_{G'}(S' \cup \{e\}) - f_{G'}(S')) \\ &= \sum_{G': \text{possible graphs}} \Pr[G'] f_{G'}(S' \cup \{e\}) - \sum_{G': \text{possible graphs}} \Pr[G'] f_{G'}(S') \\ &= f(S' \cup \{e\}) - f(S') = \Delta_e f(S') \end{aligned}$$

$$\therefore \Delta_e f(S) \geq \Delta_e f(S') \quad \square$$

Linear Programming (LP)

Input: Matrix  $A$ , vectors  $b, c$

Output: vector  $x$

Constraint:  $Ax \leq b$   $x \geq 0$

Objective Function: Maximize  $c^t \cdot x$

can be solved by

CPLEX (IBM)

Ex Input:  $A = \begin{bmatrix} 3 & 2 \\ 4 & 5 \end{bmatrix}$   $c = \begin{bmatrix} 10 \\ 12 \end{bmatrix}$   $b = \begin{bmatrix} 5 \\ 3 \end{bmatrix}$

Output:  $x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$

Constraint:  $\begin{bmatrix} 3 & 2 \\ 4 & 5 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \leq \begin{bmatrix} 5 \\ 3 \end{bmatrix}$   $\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \geq 0$

$$\begin{bmatrix} 3x_1 + 2x_2 \\ 4x_1 + 5x_2 \end{bmatrix} \leq \begin{bmatrix} 5 \\ 3 \end{bmatrix}$$

$$3x_1 + 2x_2 \leq 5$$

$$4x_1 + 5x_2 \leq 3$$

Objective Function:  $\begin{bmatrix} 10 \\ 12 \end{bmatrix}^T \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 10x_1 + 12x_2$

LP in another form

maximize

$$10x_1 + 12x_2$$

such that

$$\begin{cases} 3x_1 + 2x_2 \leq 5 & \times y_1 \\ 4x_1 + 5x_2 \leq 3 & \times y_2 \end{cases}$$

→ Primal Linear Programming

To reduce the computation time, we want to find the upper bound of the objective function

We want to merge the 2 inequalities to obtain

$$10x_1 + 12x_2 \leq \text{○}$$

objective function

Upper bound of the objective function

$$3x_1y_1 + 2x_2y_1 + 4x_1y_2 + 5x_2y_2 \leq 5y_1 + 3y_2$$

$$x_1(3y_1 + 4y_2) + x_2(2y_1 + 5y_2) \leq 5y_1 + 3y_2$$

10
12

$$3y_1 + 4y_2 = 10$$

$$2y_1 + 5y_2 = 12$$

Upper bound  $5y_1 + 3y_2$

We want to have upper bound as small as possible.

That will cut a lot of computation time in LP solver

minimize  $5y_1 + 3y_2$

such that  $3y_1 + 4y_2 \geq 10$

$2y_1 + 5y_2 \geq 12$

→ another LP

minimize  $\underbrace{[5 \ 3]}_{b^t} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = b^t \cdot y$

such that  $\begin{bmatrix} 3y_1 + 4y_2 \\ 2y_1 + 5y_2 \end{bmatrix} \geq \begin{bmatrix} 10 \\ 12 \end{bmatrix}$

$\begin{bmatrix} 3 & 4 \\ 2 & 5 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \geq \begin{bmatrix} 10 \\ 12 \end{bmatrix} \Rightarrow A^t y \geq c$

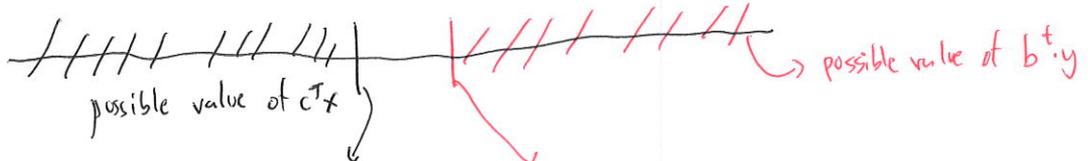
Dual Linear Programming

Primal and Dual LP

Primal  $\max c^t x$   
such that  $A \cdot x \leq b$

Dual  $\min b^t \cdot y$   
such that  $A^t \cdot y \geq c$

use for finding upper bound  
for Primal LP



we want to  
find this value  
at the primal LP

This value will be the  
best upper bound  
we can have from dual LP.