VE444 Networks 2020FA CTPP – Created by Liqin Zhang, Siyin Ma

Graph basic

maximum number of edges in 无向图

$$E_{\max} = \binom{N}{2} = \frac{N(N-1)}{2}$$

完全图的 degree=n-1

Adjacency list: large/sparse (E<<Emax (or average degree << N-1))

-average degree = 2E/N

-Degree distribution P(k): Probability randomly chosen node has degree k

-A path is sequence of nodes in which each node is linked to the next one

-Length is the number of edges in a path

-Distance between nodes is number of edges along the shortest path (asy)

-Diameter is the maximum (shortest path) distance between any nodes

-Clustering coefficient of A = pair in neighbor of A / total pairs \in [0,1]

$$C_i = \frac{2e_i}{k_i(k_i-1)}$$ k=deg. ei=#pair in nei

-giant cc is largest set where any two vertices can be joined by a path

-bridge 是一条边 删了 G 会分成两个

-local bridge 的两个端点没有相同邻居

-Span f local bridge: 删了边端点距离

-STC 三个点两个 strong 有第三边

-如果一个点有两个 s tie，他的 local bridge 一定 weak 反证法

-edge overlap = 0 -> local bridge

$$\frac{\text{与 A,B 均为邻居的节点数}}{\text{与 A,B 中至少一个为邻居的节点数}}$$

-Homophily: the tendency of individual to associate with similar others

-Homophily test: cross-attributes edges 远小于 2pq -> homophily.

-Probabilistic relational classifier

$$P(Y_i = c) = \frac{1}{|N\_i|}\sum_{(i,j)\in E} W(i,j)P(Y_j = c)$$

- $W(i,j)$ is the edge strength from i to j
- *N_i is the number of neighbors of i*

-structural balance:三点中奇数个+

	Stag	Hare	q-mix
Stag	4,4	0,3	4q, 4q+3(1-q)
Hare	3,0	3,3	3q+3(1-q), 3(1-q)
p-mix	4p+3(1-p), 4p	3(1-p), 3p+3(1-p)	

下面右面相等，右面左面相等

Pareto 最优：在不牺牲任何人的情况下，没有任何人能提升

社会最优：总和不能再提升，一定是 pareto，反之不是

Auction

First-price sealed-bid auctions

SPA: bi = vi

FPA: $s(v_i) = \left(\frac{n-1}{n}\right)v_i$

Seller revenue: n-1/n+1

Matching

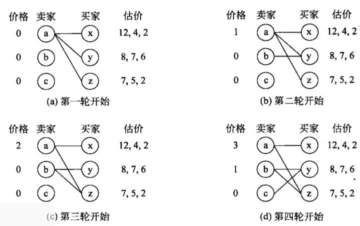
Alternating path: 在匹配与非匹配边交替的简单通路

Augmenting path: 带有非匹配端点的交替通路

1. 从右侧任意一个非匹配节点开始
2. 右找左：沿着没匹配的边扩展点
3. 左找右：沿着匹配的边扩展点

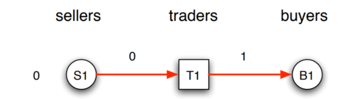
Quality of an assignment: sum of individual valuation for what they get

Optimal assignment: assignment with the maximum quality

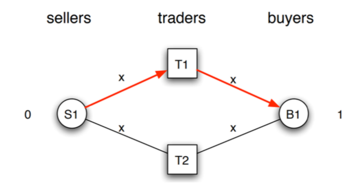


Intermediate

Monopoly 垄断：经纪人处于垄断地位



理想竞争：两个经纪人有共同的bid和ask，且唯一一个交易的T1只有0的payoff



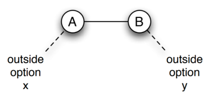
Bargain

Instability: 端点和<1

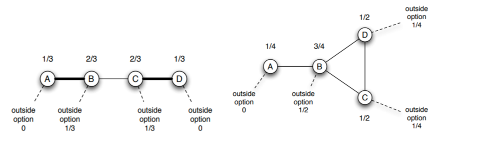
Nash bargain:

$x+\frac{1}{2}s=\frac{x+1-y}{2}$ to A,

$y+\frac{1}{2}s=\frac{y+1-x}{2}$ to B



Balanced: represent nash



两阶段版本：（有穷博弈 - 到第二周期肯定会结束）

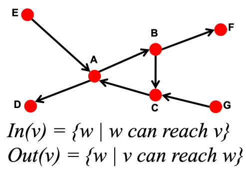
终止概率p: 在一个周期结束后谈判直接终止的概率

B可以接受的报价$(1-z,z), z = py+(1-p)(1-x)$

p越接近1，A的主动权越大；p越接近0，B的主动权越大；p越接近1/2，越接近纳什议价结果 。

无穷议价版本：（偶数长度最后议价由B给出，奇数则由A给出）

B的初始回报是$\frac{y+(1-p)(1-x)}{2-p}$,A的初始回报是$\frac{(1-p)x+1-y}{2-p}$, 当p趋近于0时即变成纳什方案的值。



In(A)={A,B,C,E,G}

Out(A)={A,B,C,D,F}

$In(v) = \{w \mid w \text{ can reach } v\}$

$Out(v) = \{w \mid v \text{ can reach } w\}$

Webpage

两种有向图的分类：

- Strongly connected: Any node can reach any node via a directed path → In(A)=Out(A)
- Directed Acyclic Graph (DAG) = scc 的 G

Strongly CC 单个节点也是 SCC，out 交 in = scc

- Every pair of nodes in S can reach each other

-There is no larger set containing S with this property

-IN: SCC 的上游，可以链接到超大 SCC 的节点，

-OUT: SCC 的下游，可以从超大 SCC 访问的节点

-Tendrils: 能够从 IN 访问但是不能链接到超大 SCC 的节点

能够链接到 OUT 但是不能从超大 SCC 访问的节点

-Tubes: 满足上面两点

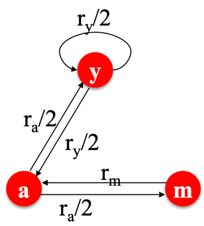
-Disconnected: 游离，和上面的图没有任何联系的点

$$r_j = \sum_{i\to j}\frac{r_i}{\mathrm{d}_i}$$

d_i … out-degree of node i

r = Mr

r_init= [1/N]



$$\begin{array}{c|ccc} & r_y & r_a & r_m \\ \hline r_y & ½ & ½ & 0 \\ r_a & ½ & 0 & 1 \\ r_m & 0 & ½ & 0 \end{array}$$

$r = M\cdot r$

"Flow" equations:

$r_y = r_y/2 + r_a/2$

$r_a = r_y/2 + r_m$

$r_m = r_a/2$

$$\begin{bmatrix} r_y \\ r_a \\ r_m \end{bmatrix} = \begin{bmatrix} ½ & ½ & 0 \\ ½ & 0 & 1 \\ 0 & ½ & 0 \end{bmatrix}\begin{bmatrix} r_y \\ r_a \\ r_m \end{bmatrix}$$

$$r_j = \sum_{i\to j}\beta\,\frac{r_i}{d_i} + (1-\beta)\frac{1}{N}$$

$$A = \beta\, M + (1-\beta)\left[\frac{1}{N}\right]_{N\times N}$$

Sponsor

-Clickthrough rate: the expected clicks per hour on an advertising slot

-Advertiser revenue per click: the expected revenue of

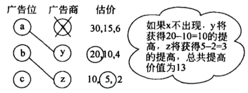
VE444 Networks 2020FA CTPP – Created by Liqin Zhang, Siyin Ma

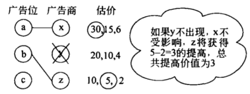
every click

-valuation: clickthrough rate * revenue per click

1. 如果知道广告价格，直接清仓

2. Vcg 真实报价是一个占优策略





3. Gsp： n 个广告商依次出价，第 i 个广告位分配给出价第 i 高的广告商，支付第 i+1 个广告商的出价

***v* chooses *A* if**

$$p > \frac{b}{a+b} = q$$

p… frac. v's nbrs. with A

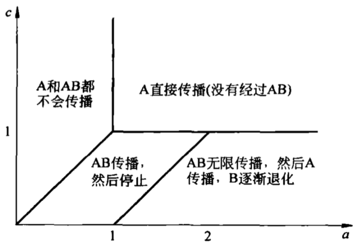
***q*… payoff threshold**

Cascade

-Cluster of density p: set of nodes fraction p neighbors in set

-If remain cluster's density greater than 1 — q, then no complete cascade.

-k-core: biggest connected subgraph where every node has at least degree k



Contagion

d: A patient meets d new people

q: With probability q > 0 he infects each of them

基本再生数：R_0 = q d

表示一个单一个体引发新病例数的期望值，也代表了疾病的持久性

$$p_h = 1 - \underbrace{(1 - q \cdot p_{h-1})^d}_{\text{No infected node at depth } h \text{ from the root}}$$

epidemic never dies R_0 >= 1

Susceptible $\xrightarrow{\beta}$ Infected $\xrightarrow{\delta}$ Recovered

Probability of infection: π

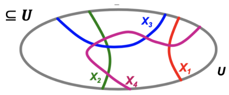
Each infected contact the number of others: k

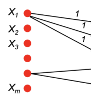
Transmission rate: β= πk, Recovery rate δ

N =S+I+R

- $\frac{dS}{dt} = -\frac{\beta S I}{N}$
- $\frac{dI}{dt} = \frac{\beta S I}{N} - \delta I$
- $\frac{dR}{dt} = \delta I$
- $R_0 = \frac{\beta}{\delta} > 1$, 每个个体期望传播个数

群体免疫 herd immunity threshold: { 1-1/R0 }





Influence Maximization

--Linear Threshold Model

-A node v has random threshold $\theta_v \in [0,1]$

-A node v is influenced by neighbor w with weight b_v,w

A node v becomes active when

$$\sum_{w \text{ active neighbor of } v} b_{v,w} \geq \theta_v$$

--Independent Cascade Model

-最初有 S 集合中的点为 active， 每边有概率 p{v,w}，每个点以该概率激活它的 out-neighbor,

-Most influential set of size k: set S of k nodes producing largest expected cascade size f(S)

-NP-hard: as hard as set cover

-Greedy Hill Climbing

At each iteration i activate the node with largest marginal gain:

$$\max_u f(S_{i-1} \cup \{u\})$$

Node representation

$$\underbrace{\text{similarity}(u,v)}_{\text{in the original network}} \approx \underbrace{\mathbf{z}_v^\top \mathbf{z}_u}_{\text{Similarity of the embedding}}$$

Shallow encoding

-Random walk: z_v^Tz_u=probability that u and v co-occur on a random walk over the network

- ppxOptimize x

- Softmax $\max_{z} \sum_{u \in V} \log P(N_R(u) | z_u)$

$$P(v|\mathbf{z}_u) = \frac{\exp(\mathbf{z}_u^\top \mathbf{z}_v)}{\sum_{n \in V} \exp(\mathbf{z}_u^\top \mathbf{z}_n)}$$

- Minimize L

$$\mathcal{L} = \underbrace{\sum_{u \in V}}_{\text{sum over all nodes } u} \underbrace{\sum_{v \in N_R(u)}}_{\text{sum over nodes } v \text{ seen on random walks starting from } u} -\log \underbrace{\left(\frac{\exp(\mathbf{z}_u^\top \mathbf{z}_v)}{\sum_{n \in V} \exp(\mathbf{z}_u^\top \mathbf{z}_n)}\right)}_{\text{predicted probability of } u \text{ and } v \text{ co-occuring on random walk}}$$

Algorithm 1 DeepWalk(G, w, d, γ, t)

Input: graph $G(V, E)$
window size w
embedding size d
walks per vertex γ
walk length t
Output: matrix of vertex representations $\Phi \in \mathbb{R}^{|V| \times d}$
Initialization: Sample Φ from $\mathcal{U}^{|V| \times d}$
Build a binary Tree T from V
for $i = 0$ to γ **do**
  $\mathcal{O} = \text{Shuffle}(V)$
  for each $v_i \in \mathcal{O}$ **do**
    $\mathcal{W}_{v_i} = RandomWalk(G, v_i, t)$
    SkipGram(Φ, $\mathcal{W}_{v_i}$, w)
  end for
end for

Algorithm 2 SkipGram(Φ, $\mathcal{W}_{v_i}$, w)

for each $v_j \in \mathcal{W}_{v_i}$ **do**
  for each $u_k \in \mathcal{W}_{v_i}[j - w : j + w]$ **do**
    $J(\Phi) = -\log \Pr(u_k \mid \Phi(v_j))$
    $\Phi = \Phi - \alpha * \frac{\partial J}{\partial \Phi}$
  end for
end for

Deep encoder

Deep encoder

$$\mathbf{h}_v^0 = \mathbf{x}_v$$

Initial 0-th layer embeddings are equal to node features

$$\mathbf{h}_v^k = \sigma\left(\mathbf{W}_k \sum_{u \in N(v)} \frac{\mathbf{h}_u^{k-1}}{|N(v)|} + \mathbf{B}_k \mathbf{h}_v^{k-1}\right), \quad \forall k \in \{1, \ldots, K\}$$

$$\mathbf{z}_v = \mathbf{h}_v^K$$

Previous layer embedding of *v*

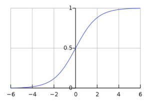
Average of neighbor's previous layer embeddings

Embedding after K layers of neighborhood aggregation

Non-linearity (e.g., ReLU)

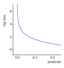
- Sigmoid $S(x) = \frac{1}{1+e^{-x}} = \frac{e^x}{e^x+1}.$
- Rectified linear activation function:

$$f(x) = x^+ = \max(0, x)$$



Cross-entropy loss(supervised)

- Loss function: $-\sum_{C=1}^{M} y_{o,c} \log(p_{o,c})$
  - M: number of classes.
  - Y: binary indicator of the true label
  - P: predicted probability of observation o is of class c



Price of Anarchy(POA): the ratio between the system performance with strategic players and the best-possible system performance

PinSage

$$\mathcal{L} = \sum_{(u,v) \in \mathcal{D}} \max(0, -\mathbf{z}_u^\top \mathbf{z}_v + \mathbf{z}_u^\top \mathbf{z}_n + \Delta)$$

set of training pairs from user logs

"positive"/true training pair

"negative" example

"margin" (i.e., how much larger positive pair similarity should be compared to negative)