

★ Network at time t : N_t

Evolving Network : New links at $t+1$ (People You May Know)

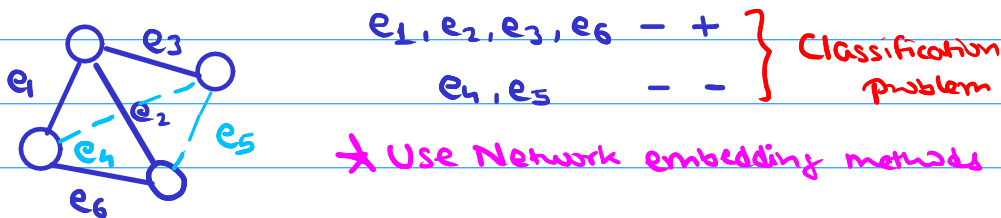
Static Network : Missing links at t (Protein)

★ Using the topological Structure

① Local Measure

② Global Measure

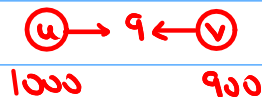
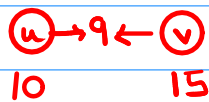
★ Machine Learning



→ Topological Structure

① Local Measures

Ⓐ Common Neighbours (Friend of Friends)



Doesn't account the ratio of common friends & total friends

★ $N(u) \rightarrow$ Neighbours of u

★ $N(v) \rightarrow$ Neighbours of v

Ⓑ Jaccard Coefficient

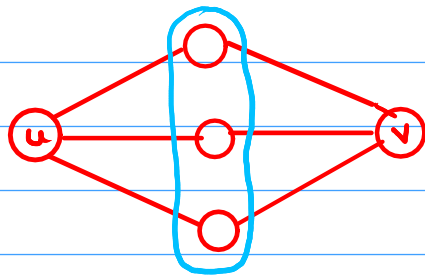
$$J(u, v) = \frac{|N(u) \cap N(v)|}{|N(u) \cup N(v)|}$$

Ⓒ Preferential Attachment

$$PT(u) \sim \text{degree}(u) \approx P_u$$

$$PT(u, v) = P_u \cdot P_v \approx \text{degree}(u) \cdot \text{degree}(v)$$

(d) Adamic Adar



$$\star AAC(u,v) = \sum_{x \in N(u) \cap N(v)} \frac{1}{\log |N(x)|}$$

$$\textcircled{c} \text{ Resource Allocation} = \sum_{x \in N(u) \cap N(v)} \frac{1}{|N(x)|}$$

② Global Measure

① Katz link Prediction (Better than Local measures)

$$\text{sim}(u,v) = \sum_{l=1}^{\infty} \beta^l |\text{path}_{u,v}^l|$$

Similarity b.w. u & v

$$\star \beta < 1$$

\Rightarrow lower weightage to higher path lengths

$$\text{sim}(u,v) = \beta |\text{path}_{u,v}^1| + \beta^2 |\text{path}_{u,v}^2| + \dots$$

$$= \beta A[u][v] + \beta^2 A^2[u][v] + \dots$$

$\star A$: Adjacency matrix

② Rooted PageRank

\star Feed u as root & get vector URP

v as root & get vector VRP

$$URP + VRP = \text{sim}(u,v)$$

\rightarrow Sim Rank

\rightarrow Hit Rank

→ Machine Learning

The main trick is to convert $e_i \langle u, v \rangle$ to a vector

① By User explicit features

a) $\deg u, \deg v$

b) Avg path length, len of shortest path, #paths

c) CC, IC, AA, RA (Local Link prediction)

② Using Deep learning

→ Node embedding → PCA, SVD, DeepWalk, Node2Vec

