Machine Learning with Graphs (MLG)

Link Prediction on Attributed Graphs

What if nodes contain user profiles?

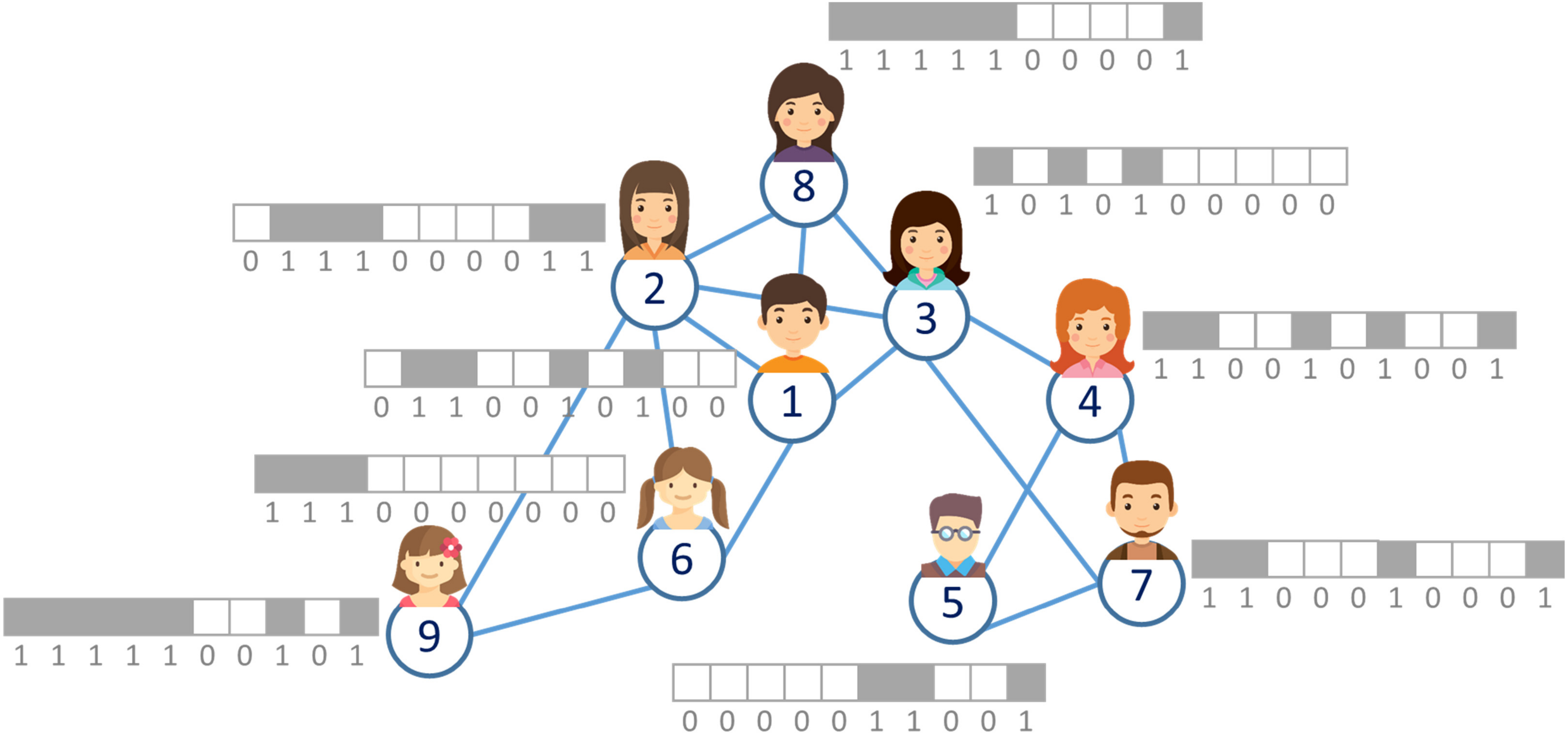
Cheng-Te Li (李政德)

Institute of Data Science

National Cheng Kung University

chengte@mail.ncku.edu.tw

Attributed Graphs

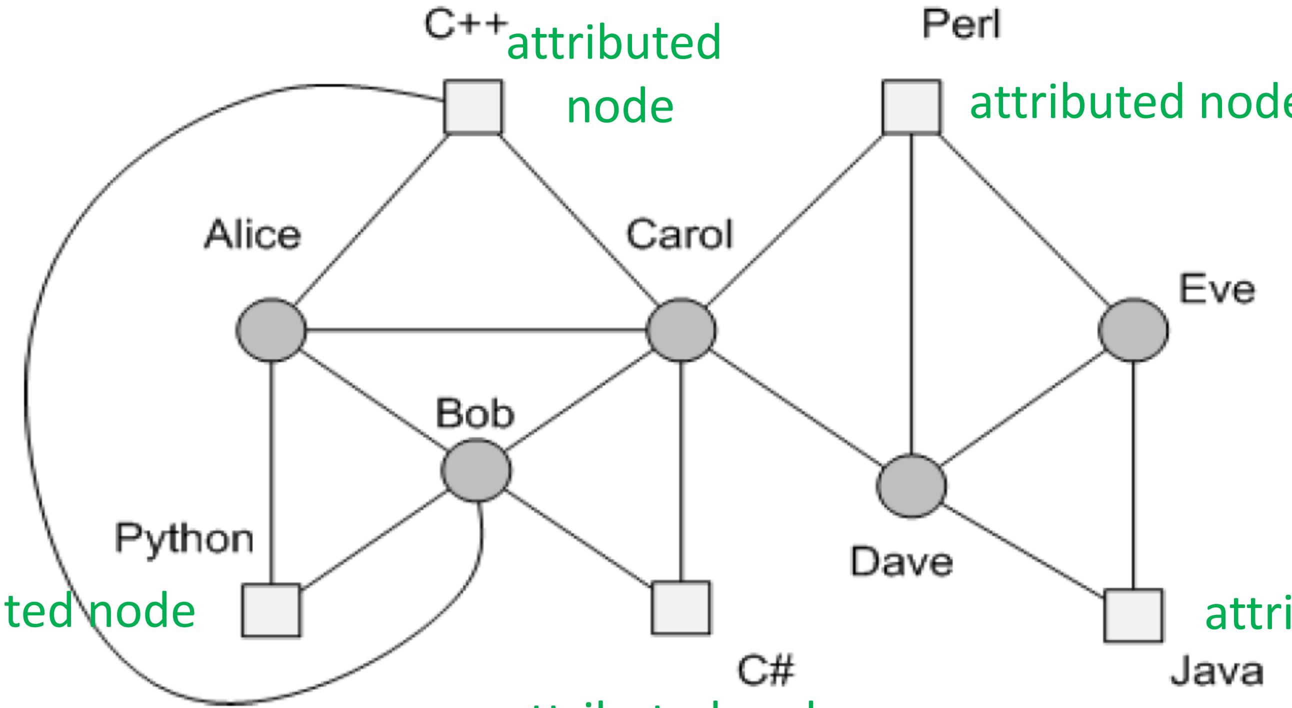


Link Prediction on AG

- Given a social graph $G = (V, E)$
  - V is the set of nodes (person)
  - E is the set of edges (link)
  - Each node has his/her own attributes
- Link prediction
  - Given node $v \in V$, the goal is to generate a list of other nodes in $u \in V$ $(u \neq v)$, in which the list is ranked by link potential

Enhanced Graph Construction

User	Attributes	Friends
Alice	“c++”, “python”	Bob, Carol
Bob	“c++”, “c#”, “python”	Alice, Carol
Carol	“c++”, “c#”, “perl”	Alice, Bob, Dave
Dave	“java”, “perl”	Carol, Eve
Eve	“java”, “perl”	Dave



attributed node

attributed node

attributed node

attributed node

attributed node

Enhanced Graph Construction (cont.)

- Assigned weights to each attribute **uniformly**
  - Use uniform weighing schema

$$w(a,p) = \frac{1}{|N_p(a)|}$$

$N_p(a)$: the set of person nodes connected to attribute node a

  - Use $\lambda \in [0,1]$ to control the trade-off between attribute and graph structure

$$w(p,a) = \begin{cases} \frac{\lambda}{|N_a(p)|}, & \text{if } |N_a(p)| > 0 \text{ and } |N_p(p)| > 0 \\ \frac{1}{|N_a(p)|}, & \text{if } |N_a(p)| > 0 \text{ and } |N_p(p)| = 0 \\ 0, & \text{otherwise} \end{cases}$$

$N_a(p)$, $N_p(p)$: the set of attribute/person nodes connected to node p

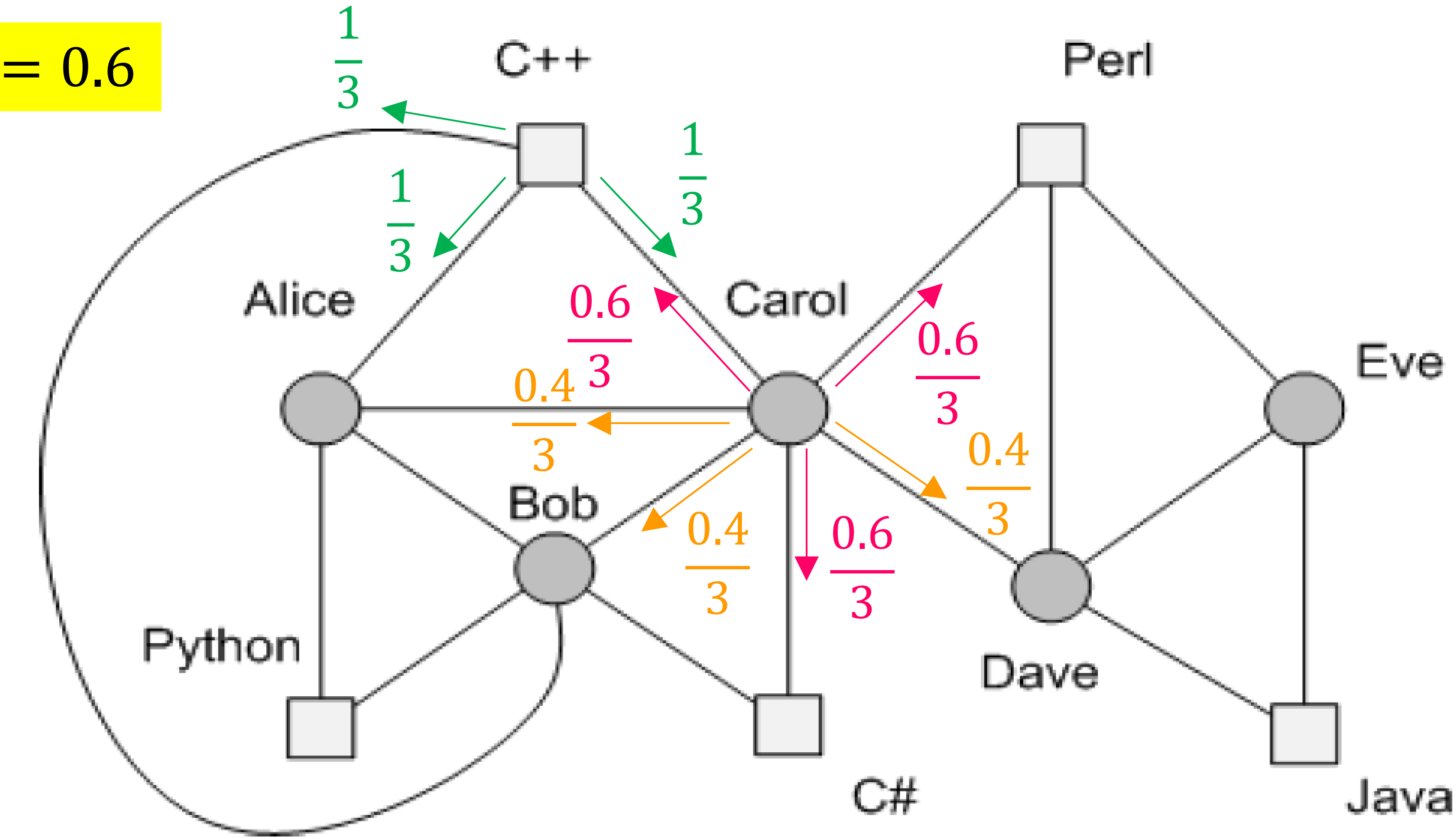
$$w(p,p') = \begin{cases} \frac{1-\lambda}{|N_p(p)|}, & \text{if } |N_p(p)| > 0 \text{ and } |N_a(p)| > 0 \\ \frac{1}{|N_p(p)|}, & \text{if } |N_p(p)| > 0 \text{ and } |N_a(p)| = 0 \\ 0, & \text{otherwise} \end{cases}$$

Larger λ → is, the more the model uses attributes for prediction

Example of Edge Weighting

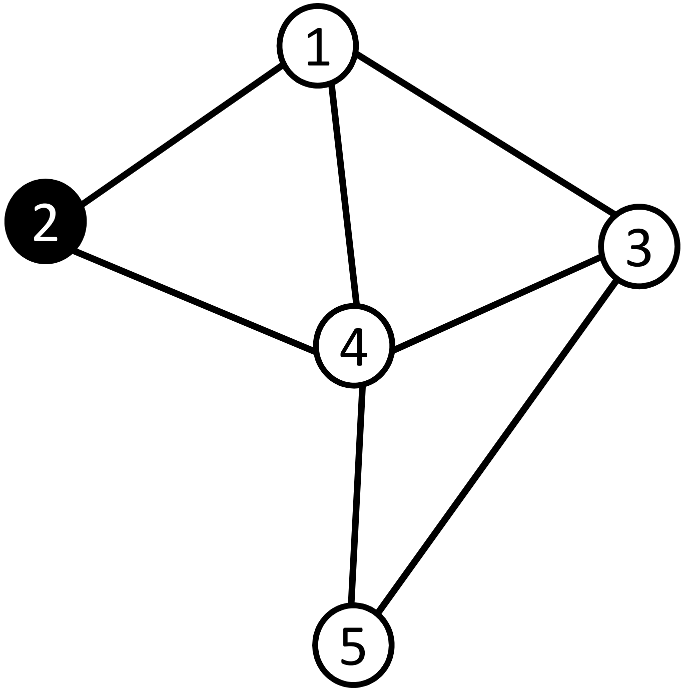
$$w(a,p) = \frac{1}{|N_p(a)|} \qquad w(p,a) = \frac{\lambda}{|N_a(p)|} \qquad w(p,p') = \frac{1-\lambda}{|N_p(p)|}$$

$\lambda = 0.6$



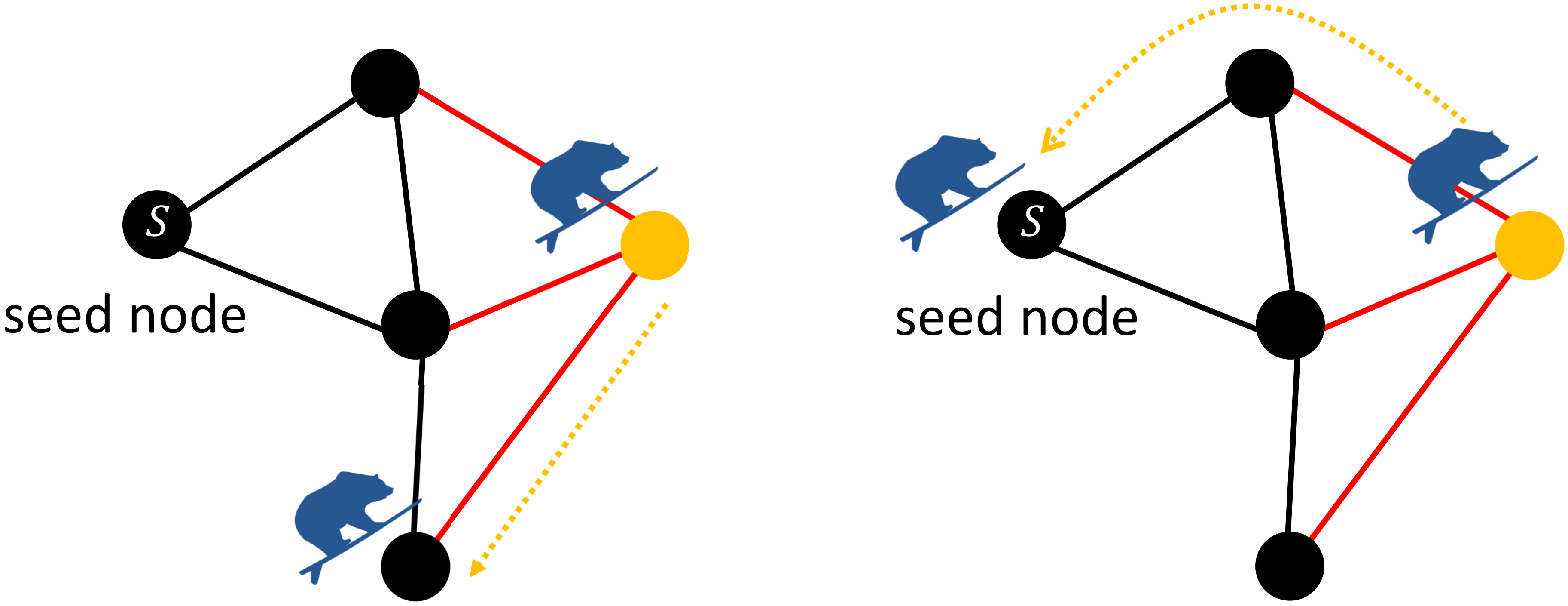
Link Potential

- How can we measure the relevance (or similarity) between two nodes in a graph?
- Example:



Random Walk with Restart (RWR)

- **Random Walk with Restart** (**RWR**) assumes a random surfer on a graph

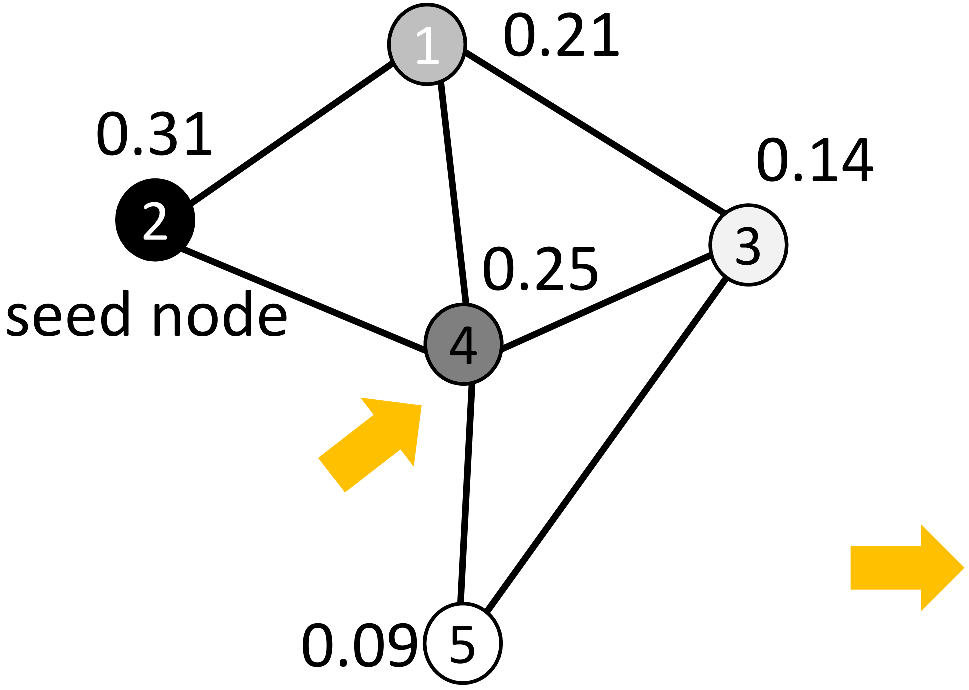


Random walk (with prob $1 - c$)

Restart (with prob c)

Random Walk with Restart (RWR)

- **RWR** computes the stationary probability that the surfer stays at each node



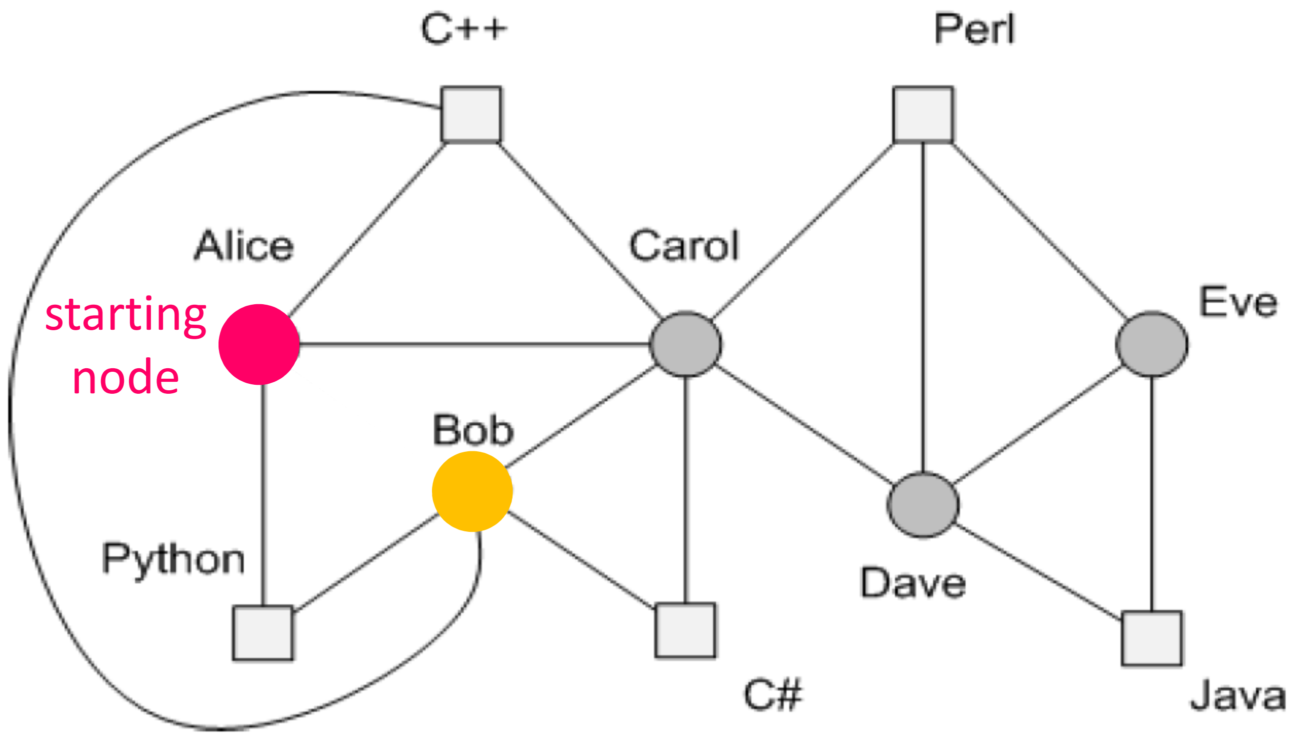
Restarting probability c = 0.2

RWR Score Vector $\vec{r}$

Node	RWR Score (relevance with node 2)
1	0.21
2	-
3	0.14
4	0.25
5	0.09

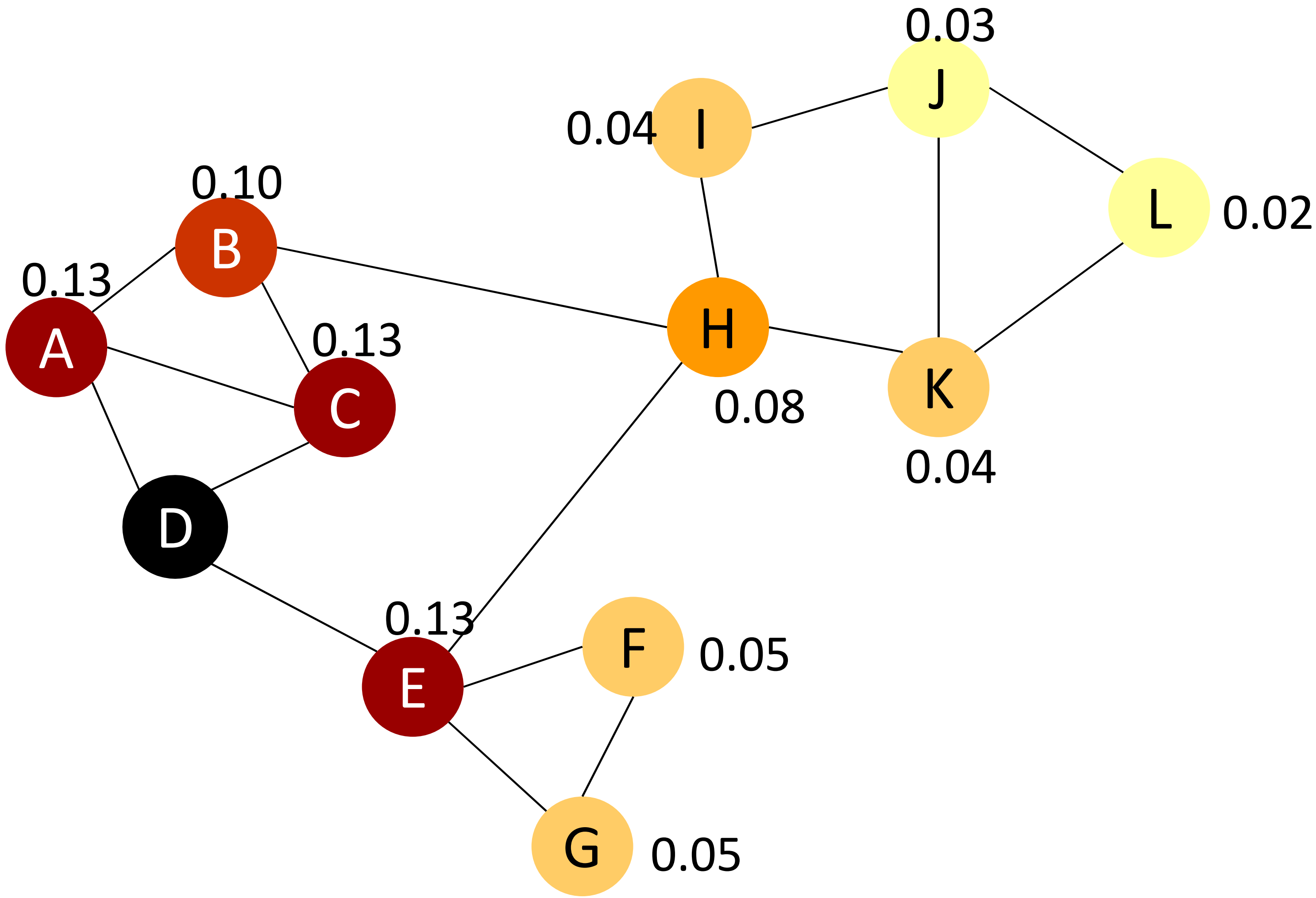
RWR Algorithm

- Random walk (RW) simulates the friendship hunting behaviors of users
- The stationary probabilities of RW starting from a given person node are considered as the link potential between person nodes

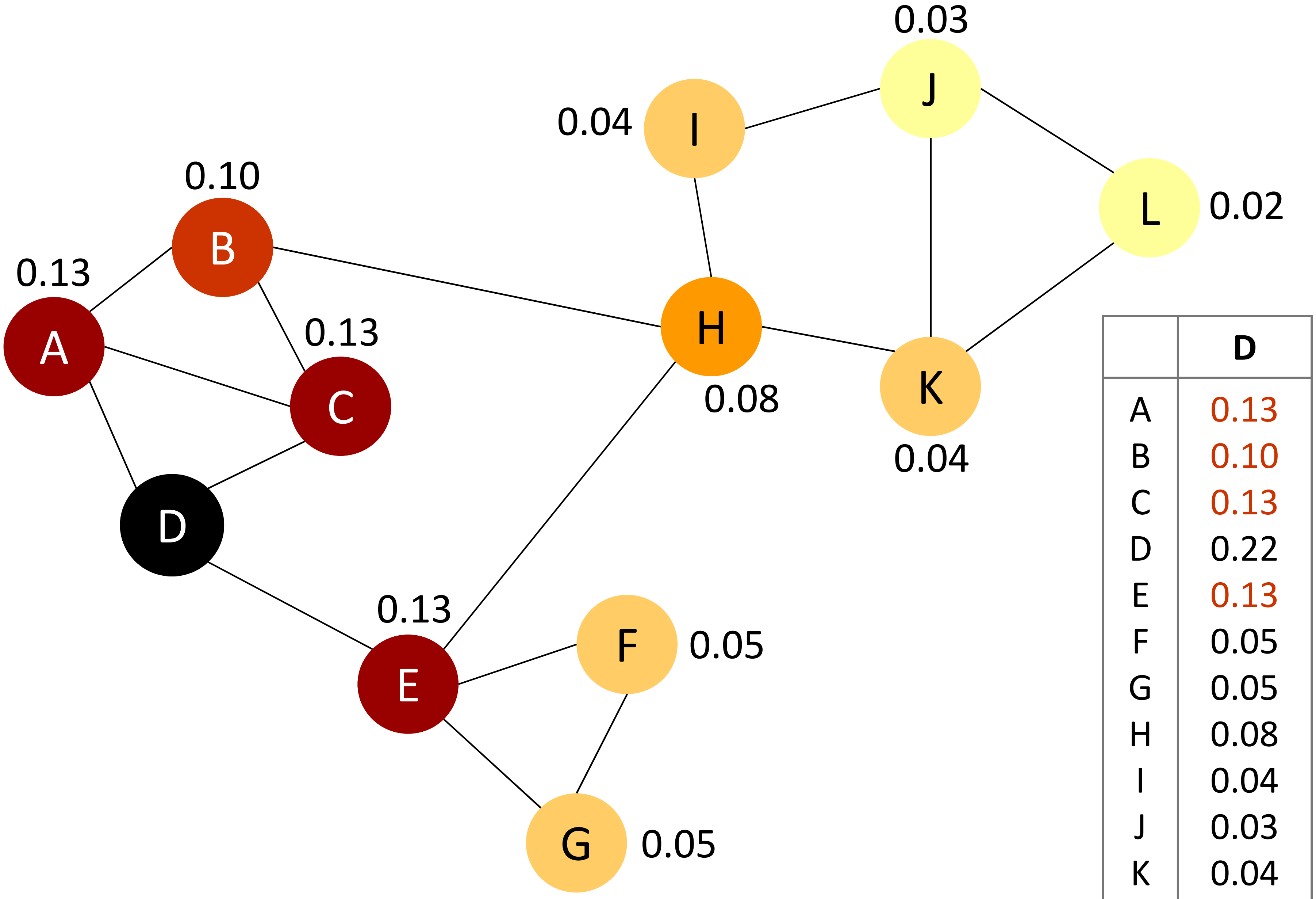


RWR Running Example

- Link potential = how easy a node reach the other node



RWR Running Example



	D
A	0.13
B	0.10
C	0.13
D	0.22
E	0.13
F	0.05
G	0.05
H	0.08
I	0.04
J	0.03
K	0.04
L	0.02

RWR Formulation

$$R = (1-\alpha)\widetilde{W}R + \alpha \cdot E$$

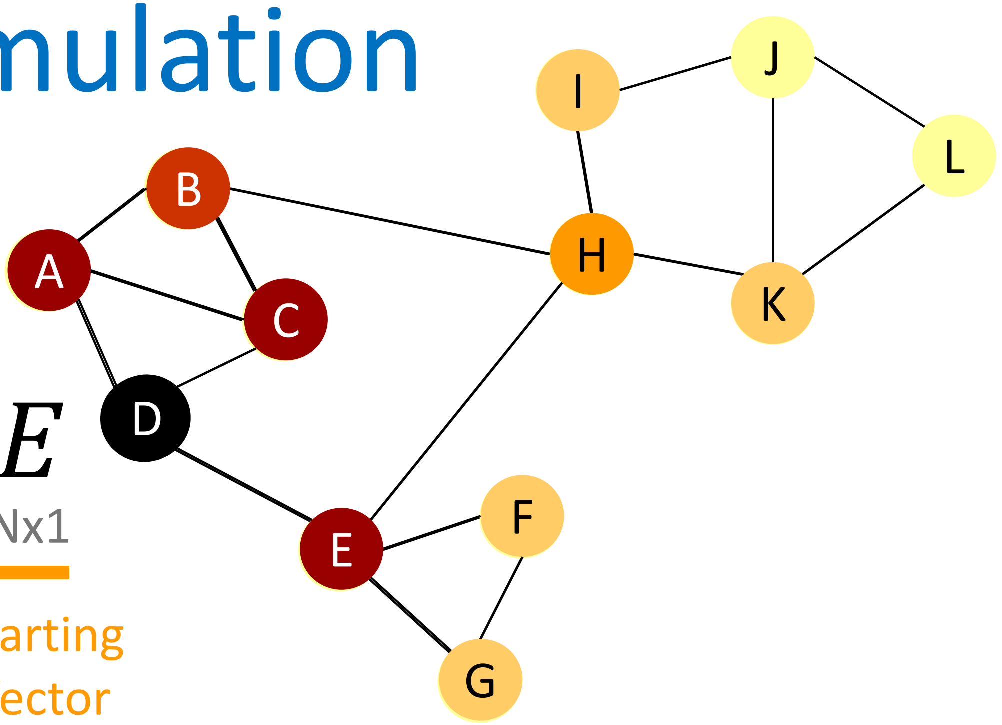
Nx1 — Score Vector

NxN — Weight Matrix

Nx1

Restarting Probability

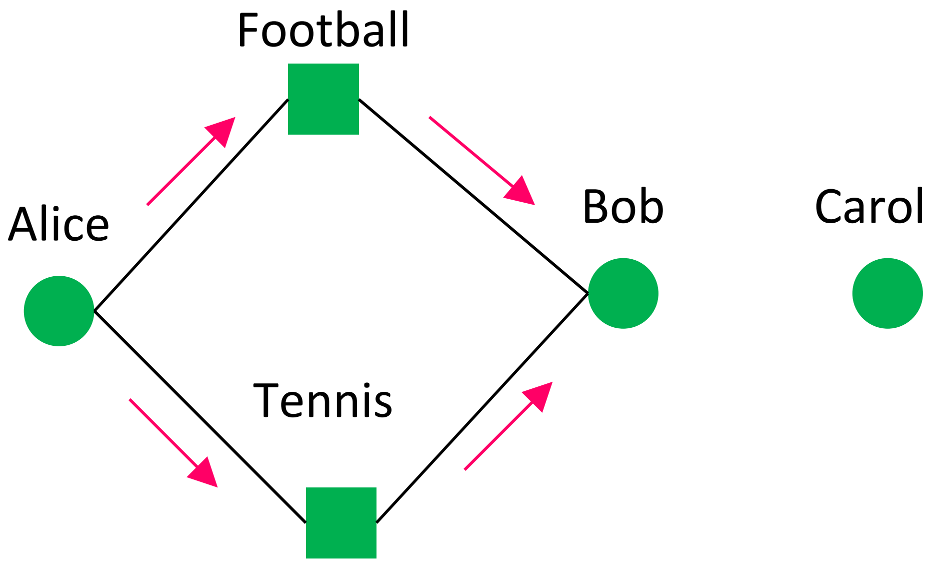
Nx1 — Starting Vector



$$\begin{pmatrix} 0.12 \\ 0.10 \\ 0.12 \\ 0.25 \\ 0.13 \\ 0.05 \\ 0.05 \\ 0.08 \\ 0.04 \\ 0.03 \\ 0.04 \\ 0.02 \end{pmatrix} = 0.9 \times \begin{pmatrix} 0 & 1/3 & 1/3 & 1/3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1/3 & 0 & 1/3 & 0 & 0 & 0 & 0 & 1/4 & 0 & 0 & 0 & 0 \\ 1/3 & 1/3 & 0 & 1/3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1/3 & 0 & 1/3 & 0 & 1/4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/3 & 0 & 1/2 & 1/2 & 1/4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/4 & 0 & 1/2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/4 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1/3 & 0 & 0 & 1/4 & 0 & 0 & 0 & 1/2 & 0 & 1/3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/4 & 0 & 1/3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/2 & 0 & 1/3 & 1/2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/4 & 0 & 1/3 & 0 & 1/2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/3 & 1/3 & 0 \end{pmatrix} \begin{pmatrix} 0.13 \\ 0.10 \\ 0.13 \\ 0.22 \\ 0.13 \\ 0.05 \\ 0.05 \\ 0.08 \\ 0.04 \\ 0.03 \\ 0.04 \\ 0.02 \end{pmatrix} + 0.1 \times \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

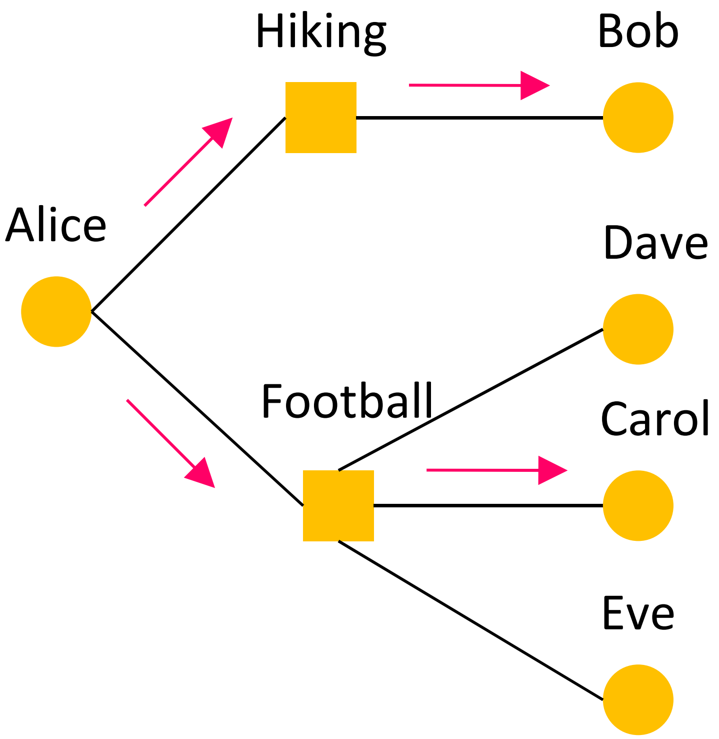
RW on Homophily

- If two persons share more attributes, the corresponding person nodes in the graph will have more connected attribute nodes in common



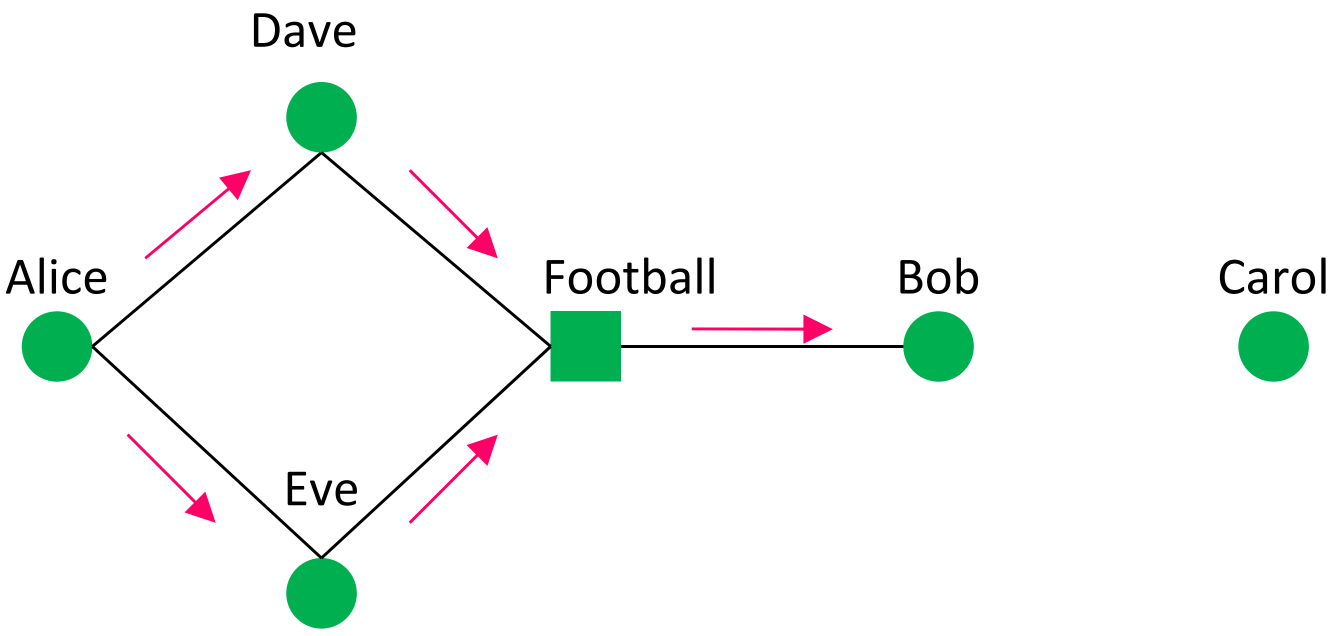
RW on Rarity

- If one attribute is rare, there are fewer outlinks for the corresponding attribute node → the weight of each outlink is larger, because there are fewer outlinks



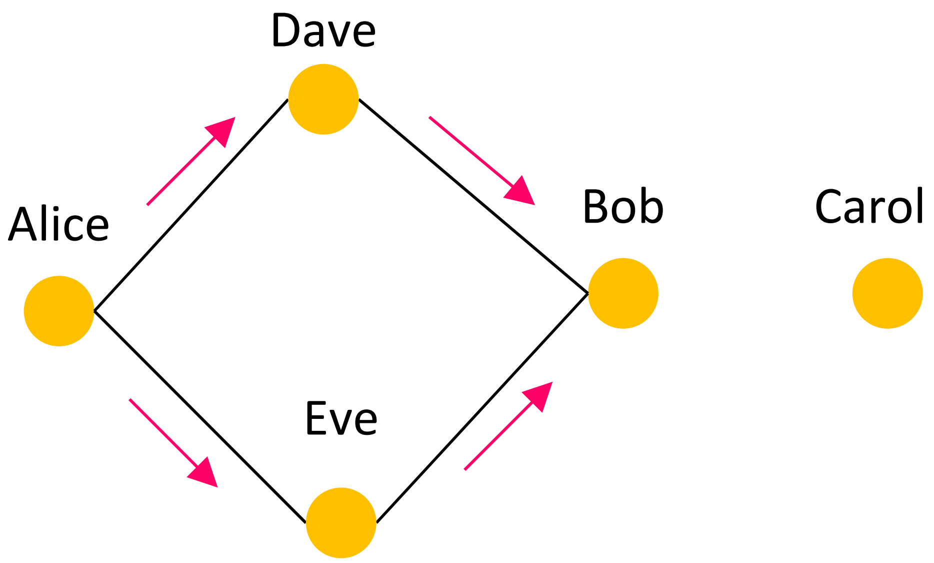
RW on Social Influence

- If one attribute is shared by many of the existing linked persons of the given person, the random walk will pass through the existing linked person nodes to this attribute node



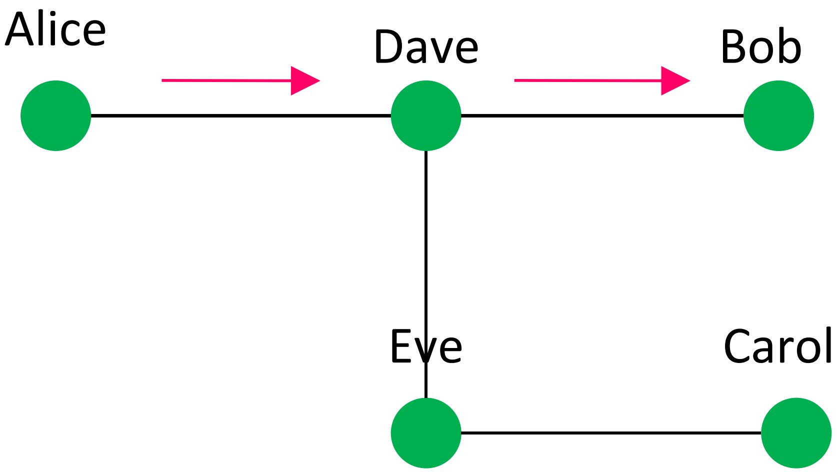
RW on Common Friends

- If two persons share many friends, these two person nodes have a large number of common neighbors in the graph



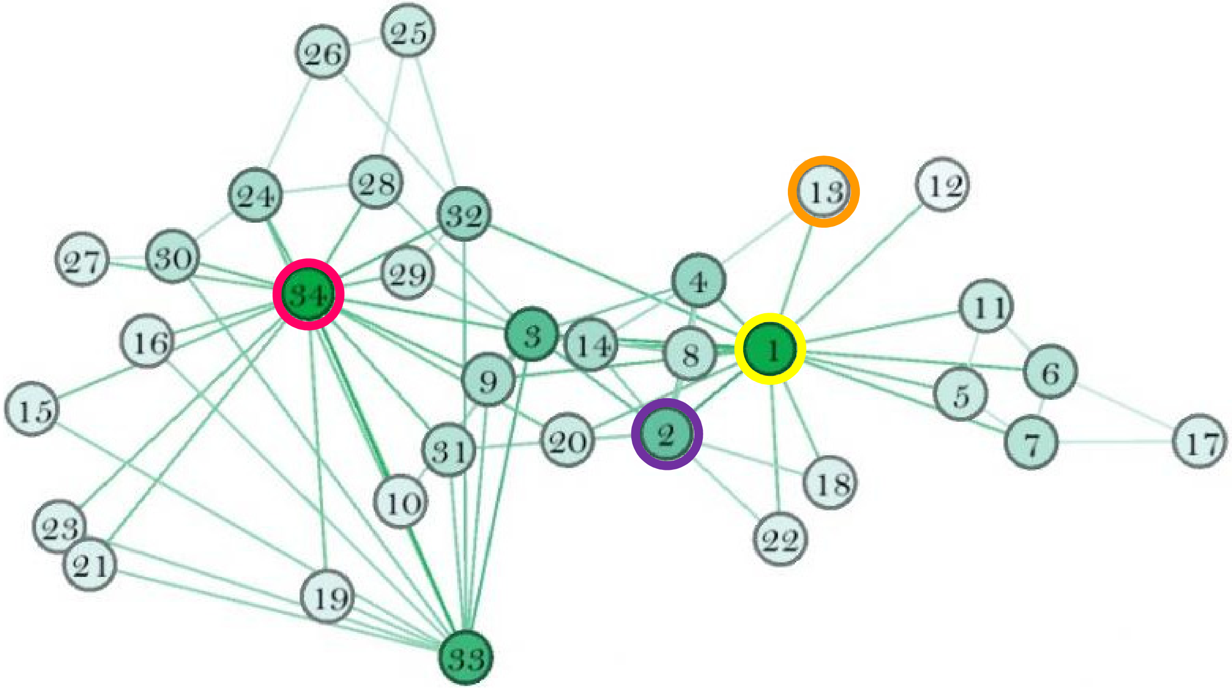
RW on Social Closeness

- If two persons are close to each other in the graph, the random walk probability from one to the other is likely to be larger than if they are far away from each other



RW on Preferential Attachment

- If a person is very popular and links to many persons, there are many in-links to the person node
  - → For a random person node in the graph it is easier to access a node with more in-links
  - E.g., Bob is very popular and has thousands of friends, but Carol has only ten friends



Edge **Re**-Weighting

- Previously we assigned weights to each attribute equally without any preference
- We should determine weights of edges $w(p, a)$ from persons to attributes based on the importance $w_p(a)$ of attribute a with regard to person p

$$w(p,a) = \begin{cases} \dfrac{\lambda w_p(a)}{\sum_{a' \in N_a(p)} w_p(a')}, & \text{if } |N_a(p)| > 0 \text{ and } \left|N_p(p)\right| > 0 \\ \dfrac{w_p(a)}{\sum_{a' \in N_a(p)} w_p(a')}, & \text{if } |N_a(p)| > 0 \text{ and } \left|N_p(p)\right| = 0 \\ 0, & \text{otherwise} \end{cases}$$

Global & Local Weighting

- GW: percentage of existing links among all the possible person pairs possessing attribute a

$$g(a) = \frac{\sum_{(u,v)\in E} e_{uv}^{a}}{\binom{n_a}{2}}$$

$e_{uv}^{a} = 1$ if persons u and v both have attribute a
n_a: number of persons having attribute a

- LW: the more the number of friends sharing the attribute a, the more important the attribute is for the person

$$l_p(a) = \sum_{p'\in N_p(p)} A(p', a)$$

$A(p, a) = 1$ if persons p has attribute a
n_a: number of persons having attribute a

- Mixed weighting

$$w_p(a) = g(a) \times l_p(a)$$

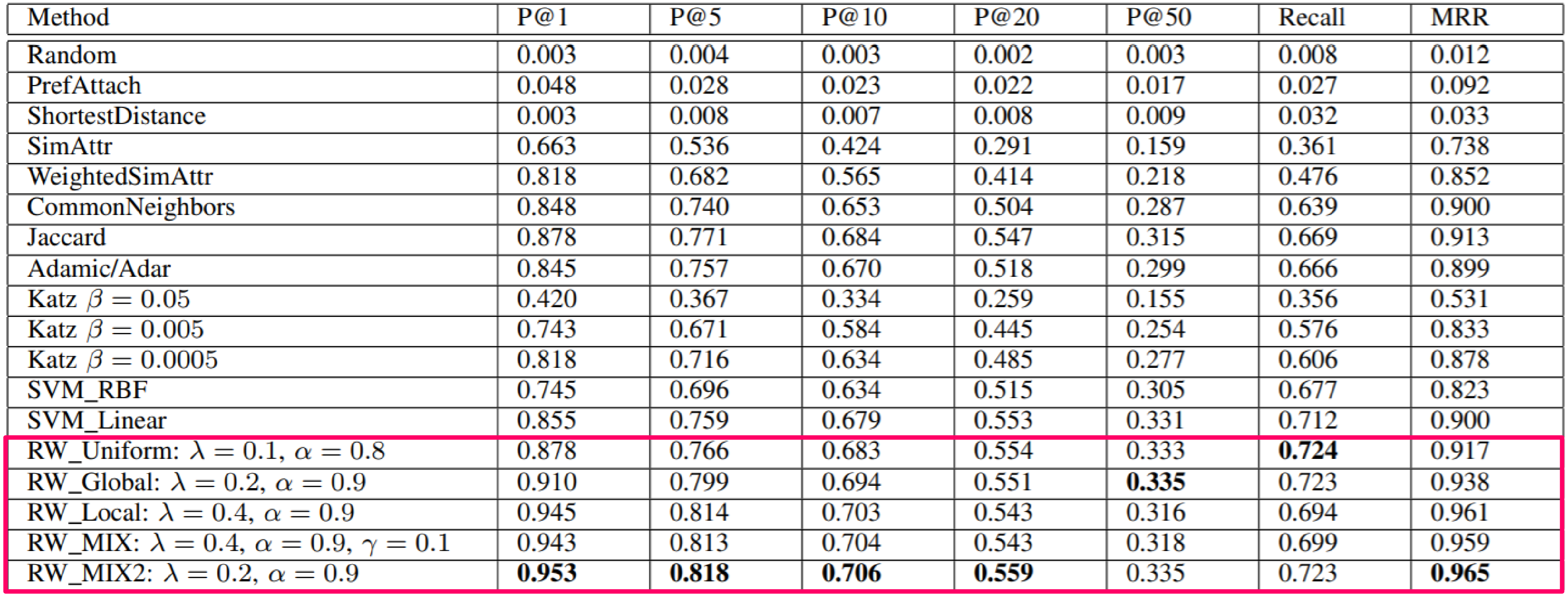
Mix1

$$w_p(a) = \gamma \frac{g(a)}{\sum_{a'\in N_a(p)} g(a')} + (1-\gamma) \frac{l_p(a)}{\sum_{a'\in N_a(p)} l_p(a')}$$

Mix2

Performance Comparison

- RW (unsupervised) > SVM (supervised) > features
- RW(MIX1) ≈ RW(MIX2) > RW(other)
- Hyperparameters λ, α, γ are quite sensitive



Method	P@1	P@5	P@10	P@20	P@50	Recall	MRR
Random	0.003	0.004	0.003	0.002	0.003	0.008	0.012
PrefAttach	0.048	0.028	0.023	0.022	0.017	0.027	0.092
ShortestDistance	0.003	0.008	0.007	0.008	0.009	0.032	0.033
SimAttr	0.663	0.536	0.424	0.291	0.159	0.361	0.738
WeightedSimAttr	0.818	0.682	0.565	0.414	0.218	0.476	0.852
CommonNeighbors	0.848	0.740	0.653	0.504	0.287	0.639	0.900
Jaccard	0.878	0.771	0.684	0.547	0.315	0.669	0.913
Adamic/Adar	0.845	0.757	0.670	0.518	0.299	0.666	0.899
Katz $\beta = 0.05$	0.420	0.367	0.334	0.259	0.155	0.356	0.531
Katz $\beta = 0.005$	0.743	0.671	0.584	0.445	0.254	0.576	0.833
Katz $\beta = 0.0005$	0.818	0.716	0.634	0.485	0.277	0.606	0.878
SVM_RBF	0.745	0.696	0.634	0.515	0.305	0.677	0.823
SVM_Linear	0.855	0.759	0.679	0.553	0.331	0.712	0.900
RW_Uniform: $\lambda = 0.1$, $\alpha = 0.8$	0.878	0.766	0.683	0.554	0.333	**0.724**	0.917
RW_Global: $\lambda = 0.2$, $\alpha = 0.9$	0.910	0.799	0.694	0.551	**0.335**	0.723	0.938
RW_Local: $\lambda = 0.4$, $\alpha = 0.9$	0.945	0.814	0.703	0.543	0.316	0.694	0.961
RW_MIX: $\lambda = 0.4$, $\alpha = 0.9$, $\gamma = 0.1$	0.943	0.813	0.704	0.543	0.318	0.699	0.959
RW_MIX2: $\lambda = 0.2$, $\alpha = 0.9$	**0.953**	**0.818**	**0.706**	**0.559**	0.335	0.723	**0.965**

Opportunities

- Attributes may be correlated with each other
  - → How to automatically model their correlation?
- The enhanced graph is based on categorical attribute
  - → How to handle numerical attributes?
- Link online and offline social world
  - → What if users have check-ins in geography?
- Links from 0 to 1 vs. links from 1 to 0
  - → Can we predict “unfriend” links?