Enhance Diversity in Prestige

Refer : http://www-personal.umich.edu/~qmei/pub/kdd10-divrank.pdf

Introduction

Centrality

- Centroid of clusters
- Highly connected in social Nets
- Support Vectors in SVM

Diversity

- Distinct information
- Coverage(Widespread)
- Variance, Entropy

Introduction

Centrality Metrics

- Degree
- Closeness
- Betweenness
- EigenVector (RandomWalk)

Enhance Diversity

- Greedy Vertex Selection and Weight updates
- Cluster the nodes (How many ?)

Vertex Reinforced Random Walk

$$p_0(u,v) = \begin{cases} \alpha \frac{w(u,v)}{\text{deg}(u)}, & \text{if } u \neq v \\ 1-\alpha, & \text{otherwise} \end{cases}$$

$$p_T(u,v) = (1-\lambda)p^*(v) + \lambda \frac{p_0(u,v)N_T(v)}{D_T(u)}$$

$$D_T(u) = \sum_{v \in V} p_0(u,v)N_T(v)$$

$$E[N_{T+1}(v)] = E[N_T(v)] + p_{T+1}(v)$$

$$p_T(v) = \sum_{(u,v) \in E} p(u,v)p_{T-1}(u)$$ [Standard Random Walk]

$$p_T(v) = \sum_{(u,v) \in E} p_T(u,v)p_{T-1}(u)$$

CumDivRank

$$E[N_T(v)] \propto \sum_{t=0}^{t=T} p_t(v)$$

PointwiseDivRank

$$E[N_T(v)] \propto p_T(v)$$

Cumulative DivRank Implementation

$$p_T(u, v) = (1 - \lambda)p^*(v) + \lambda \frac{p_0(u, v)N_T(v)}{D_T(u)}$$

$$D_T(u) = \sum_{v \in V} p_0(u, v)N_T(v)$$

$$p_0(u, v) = \begin{cases} \alpha \frac{w(u, v)}{\text{deg}(u)}, & \text{if } u \neq v \\ 1 - \alpha, & \text{otherwise} \end{cases}$$

$$E[N_{T+1}(v)] = E[N_T(v)] + p_{T+1}(v)$$

$$p_T(v) = \sum_{(u,v) \in E} p_T(u, v)p_{T-1}(u)$$

$$E[N_T(v)] \propto \sum_{t=0}^{t=T} p_t(v)$$

$$p_T(u, v) = (1 - \lambda)p^*(v) + \lambda \frac{p_0(u, v) \sum_{t=0}^{t=T} p_t(v)}{D_T(u)}$$

$$D_T(u) = \sum_{v \in V} p_0(u, v) \sum_{t=0}^{t=T} p_t(v)$$

Both the terms need to be normalised,
to handle overflow

$$N_T(v) = \frac{\sum_{t=0}^{t=T} p_t(v)}{\sum_{u \in V} \sum_{t=0}^{t=T} p_t(u)}$$

Cumulative DivRank Implementation

$$p_0(u,v) = \begin{cases} \alpha \dfrac{w(u,v)}{\text{deg}(u)}, & \text{if } u \neq v \\ 1-\alpha, & \text{otherwise} \end{cases}$$

$$p_T(u,v) = (1-\lambda)p^*(v) + \lambda \frac{p_0(u,v)\sum_{t=0}^{t=T} p_t(v)}{D_T(u)}$$

$$D_T(u) = \sum_{v \in V} p_0(u,v) \sum_{t=0}^{t=T} p_t(v)$$

Both the terms need to be normalised, to handle overflow -> $N_t(v)$

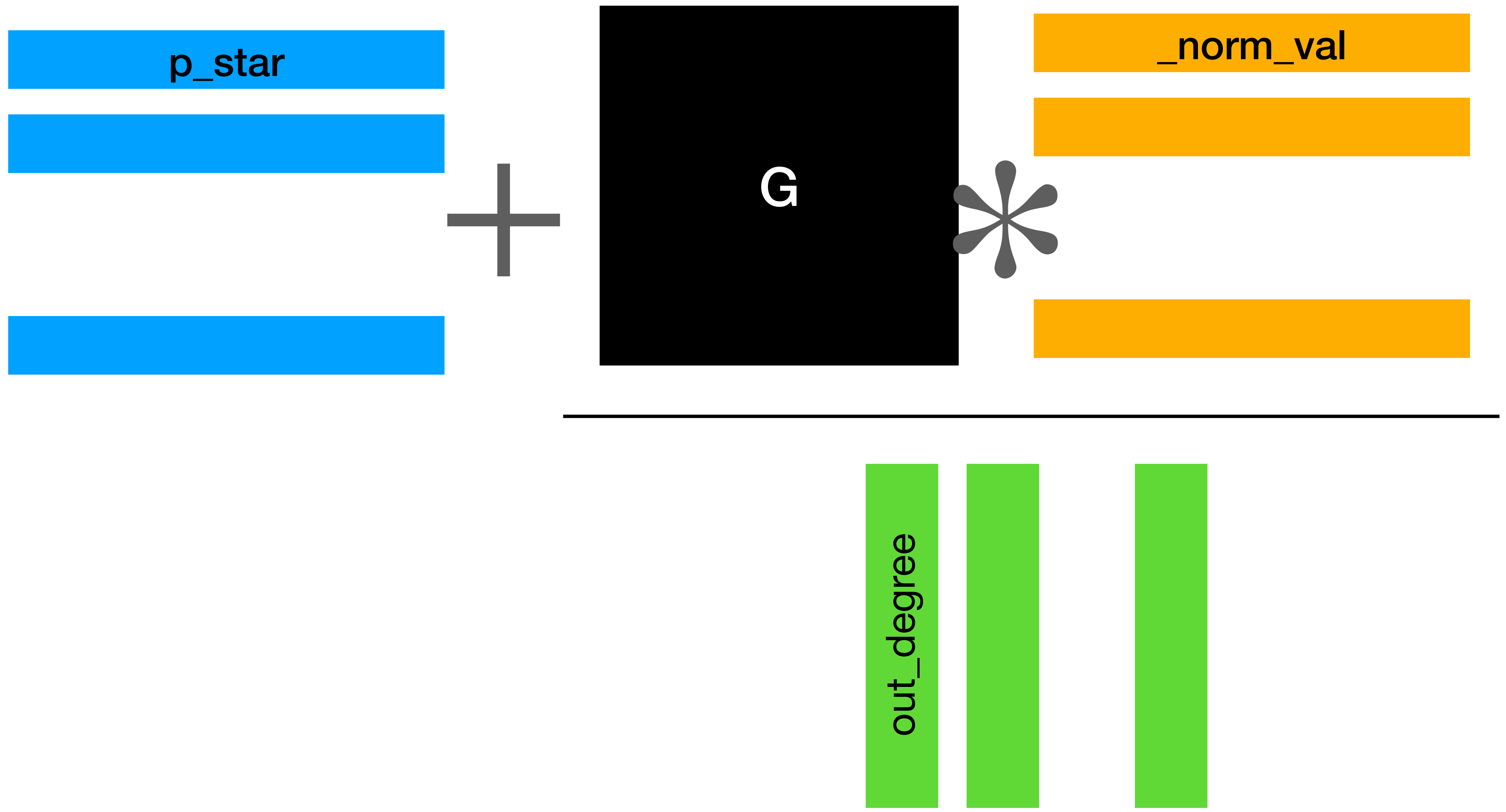
$$p_T(v) = \sum_{(u,v)\in E} p_T(u,v)p_{T-1}(u)$$

```
1. out_degree = G @ _norm_val # nx1

2. G_new = (1-d)*p_star.reshape(1,-1) +
          (d * (G * _norm_val.reshape(1,-1))/ out_degree)

3. x_next = (x_prev.T @ G_new).T # nx1
```

- $P \longrightarrow$ G : nxn array
- $N_t \longrightarrow$ _norm_val : nx1 array
- $D_t \longrightarrow$ out_degree <- D_t : nx1 array
- $\lambda \longrightarrow$ d



Cumulative DivRank Implementation

Scalability

- Matrix Implementation
- Dict of Dict

Metrics

Author/Movie Network

- Density as reverse of diversity
- Country(movie) Coverage
- Impact

Text Summarisation

- TF-Idf vectorise
- Rougue Score