CS 768: Learning With Graphs **Autumn 2020-2021**

Lecture 10: Supervised Random Walk

Instructor: Prof. Abir De *Scribe: Arjit Jain & Samad Koita*

10.1 Supervised Link Prediction Methods

1. Combining scores of different unsupervised methods
   - Computing Scores: $s_w(u, v) = w_{AA}AA(u, v) + w_{CN}CN(u, v) + b$
   - Learning Weights: SVM/Ranking loss on scores
2. Supervised Random Walk
3. Collaborative Filtering based approaches

10.2 Precursor: Random Walks on Graphs

10.2.1 Link Prediction with Random Walks

Consider a source node $s \in V$ in the Graph $G = (V, E)$. Let $D = (d_1, d_2, \ldots, d_n)$ be the candidate neighbours (or destination nodes) of node s. We assume that s and d are likely to be connected in the future if:

- The structure around s and d is rich - their surrounding region has a lot of edges.
- There is some affinity between s and d

One way to judge how rich the structure is around s and d is using random walks. Let $count(s, d)$ is the number of times d is reached in a random walk starting at s. Let $p(s, d)$ be the probability of reaching d from a random walk starting at s. We define $score(s, d)$ as:

$$score(s, d) \propto count(s, d) \propto p(s, d)$$

Once we obtain the scores for each $d_i \in D$, we can evaluate this approach for link prediction using MAP or Average Precision.

10.2.2 Constructing a Random Walk

Consider the graph $G = (V, E)$. Let A be the adjacency matrix of the graph. $A_{ij} = 1$ if vertex v_i is connected to v_j, and 0 otherwise. How to perform a random walk on this graph? To do this we define the transition matrix $\mathbf{P}$.

Transition Matrix: $\mathbf{P}$ is said to be the transition matrix of a graph when $\forall i, j$ P_{ij} denotes the probability that v_i follows v_j in a random walk on the graph. Since you can only move to neighbouring nodes, $(A_{ij} = 0) \implies (P_{ij} = 0)$. P is a stochastic matrix - the sum of entries in each row is 1.

Once we have this matrix $\mathbf{P}$, we can use it to computer the probability of reaching one node from the other.

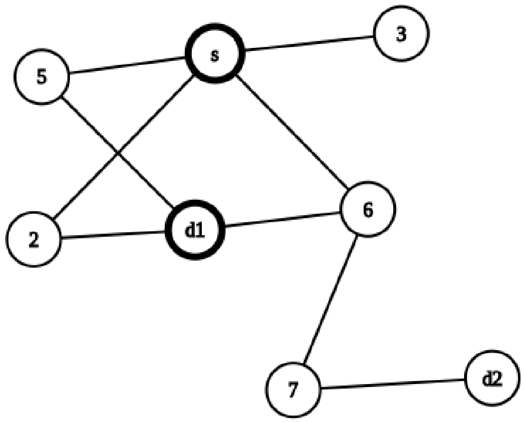

Figure 10.1: Edge (s,d_1) - High probability

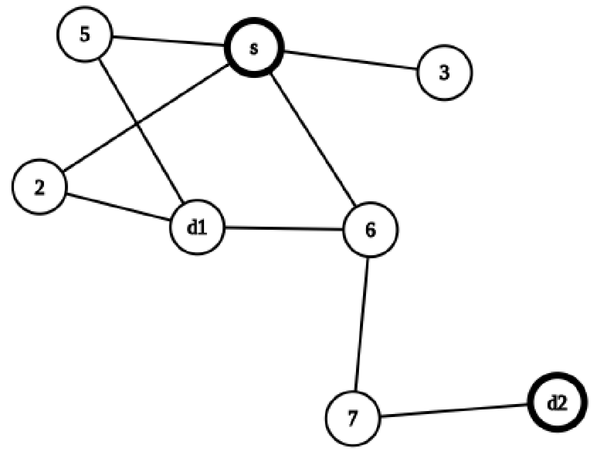

Figure 10.2: Edge (s,d_2) - Low probability

Using these probabilities, we can compute $score(s, d)$.
Traditional Random Walks assume you are equally likely to visit each one of your neighbours, and choose P in such a way that this condition is satisfied.
Supervised Random Walks get around this assumption. Using both node features, and the structure of the graph, the transition matrix P is learnt.

10.2.3 Computing Scores: PageRank

Given a transition matrix, also called weighted adjacency matrix, $\mathbf{P}$, and nodes s and d, how to compute $score(s, d)$? Let us analyse $\mathbf{P}$ a little closely. Let $P(j|i) = P_{ij}$ be the probability of the moving to node j from the node i in a random walk.

$$P_i = \begin{bmatrix} P(1|i) & P(2|i) & \dots & P(n|i) \end{bmatrix}$$

$\mathbf{P}$ is a stochastic matrix, having sum of each row as 1. Therefore,

$$\mathbf{P^M} \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} = \mathbf{P} \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}$$

From the equation above, we can see that $\lambda(\mathbf{P}) = 1$.
Given a starting state s, let $p_t = \begin{bmatrix} p_{1t} & \dots & p_{nt} \end{bmatrix}$ where p_{it} is the probability of being at node i after t steps.

$$p_t^T \mathbf{P} \;=\; p_{t+1}^T$$

Stationary Distribution: As $t \to \infty$, the probability of being at any node p_∞^T will be:

$$p_\infty^T \;=\; \lim_{t \to \infty} (p_0^T \mathbf{P}^t)$$

In most cases, this limit exists, and $p_\infty = p_{st}$, the steady state probability distribution on the graph. Clearly, $p_\infty^T = p_{st}^T\mathbf{P}$. This implies p_{st} is the left eigenvector of the eigenvalue $\lambda = 1$ of $\mathbf{P}$.

This steady state distribution, p_{st} is called the *PageRank* of the graph.

We have defined $score(s, d)$ to be proportional to the probability of reaching a node d in a random walk. Hence we define:

$$score(s, d) = p_{st}[d]$$

10.2.4 Better Metric: Personalised PageRank

An interesting ovservation is that $score(s, d) = p_{st}[d]$ is independent of the start node s. Therefore, for the purpose of link prediction, simply using PageRank is not enough to make user-specific predictions. We correct for this by introducing personalised page rank.
The idea is to introduce a restart probability, with which, a node can jump back to the source vertex s. This biases the random walks to the starting node s, and hence more localized, or personalized, compared to PageRank.

$$P^*_{ij} = (1 - \alpha)P_{ij} + \alpha I(j = s)$$

where α is the restart probability.
Hence,

$$P^*_s = (1 - \alpha)P + \alpha R_s$$

where only the entries in s^{th} column of R_s are 1, and all others are 0.
Note that

$$\mathbf{P}^* \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} = (1-\alpha)\mathbf{P} \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} + \alpha R_s \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} = (1-\alpha) \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} + \alpha \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}$$

Similar to the above derivation, we can derive p_s^T to be the solution of

$$p^T \mathbf{P}^*_{\mathbf{s}} = p^T$$

10.3 Supervised Random Walk

10.3.1 Modelling the Transition Matrix

One way to model the transition matrix P with the parameter w is as follows:

$$P_{ij} = \frac{w^T f(x_i, x_j)}{\sum_{j'} w^T f(x_i, x_{j'})}$$

where x_i, x_j are the node features of nodes i and j respectively.
Here, f should be a non-negative function, for example
$f(x, y) = ||x - y||$ (Norm function)
$f(x, y) = x \odot y$ (Element-wise dot product)
The requirement of a non negative function can be bypassed using a hinge as follows:

$$P_{ij} = \frac{[w^T f(x_i, x_j)]_+}{\sum_{j'} [w^T f(x_i, x_{j'})]_+}$$

Incorporating constraints:
For a fixed node s, we choose w as per the following objective

$$\min ||w||^2$$

such that $\forall l, d\ (s,l) \notin E, (s,d) \in E \quad p_s^T[d] = P_\infty(d|s) > P_\infty(l|s) = p_s^T[l]$
i.e. the min norm solution of w such that the probabilities, as computed by Personalized PageRank using parameter w, for neighbours of s are more than non-neighbours.
Hence, for a node s, we have $N(s) \times (|V| - N(s))$ constraints.

Note that, the optimal w can be calculated either individually for all source nodes s, or a common w can be calculated across all nodes.
In practice, the above optimization can be realised using hinge loss and some form of regularization on w. i.e.

$$\min\ \lambda ||w||^2 + \sum_{l,d\ (s,l)\notin E,(s,d)\in E} [p_s^T[l] - p_s^T[d]]_+$$

10.3.2 Recommended reading

Supervised Random Walks: Predicting and Recommending Links in Social Networks[1]

References

[1] Backstrom, L., and Leskovec, J. Supervised random walks: Predicting and recommending links in social networks. In *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining* (New York, NY, USA, 2011), WSDM '11, Association for Computing Machinery, p. 635–644.