Personalized Ranking in Signed Networks using Signed Random Walk with Restart

Abstract—**In this supplementary document, we provide description of existing methods, description of datasets and description of evaluation measurements used in the experiments on applications such as the sign prediction and the troll identification tasks.**

I. Description of Methods

SALSA [5], SR [3] and NR [3] were originally proposed to compute global rankings based on the concept of PageRank. Hence, we made a minor change in those methods to compute personalized rankings by considering only one seed node, not all nodes.

- Random Walk with Restart (RWR): We perform RWR on a given network after taking absolute edge weights. In this case, it provides only a trust ranking vector, $\mathbf{r}^+$.
- Modified Random Walk with Restart (M-RWR) [6]: M-RWR applies RWR separately on both a positive subgraph and a negative subgraph; thus, it obtains $\mathbf{r}^+$ on the positive subgraph, and $\mathbf{r}^-$ on the nagative subgraph.
- Modified Personalized SALSA (M-PSALSA) [5]: Andrew et al. made a modification on SALSA[1] by introducing the random jump into it, called Personalized SALSA (PSALSA). As similar to M-RWR, we apply PSALSA separately on both positive and negative subgraphs, and consider authorities on the positive subgraph as $\mathbf{r}^+$, and those scores on the negative subgraph as $\mathbf{r}^-$.
- Personalized Signed Spectral Rank (PSR) [3]: PSR constructs a matrix similar to Google matrix as follows:

$$\mathbf{M}_{PSR} = (1-c)\mathbf{D}^{-1}\mathbf{A}^\top + c\mathbf{e}_s\mathbf{1}^\top$$

  where $\mathbf{e}_s$ is the s-th unit vector. Then, PSR computes the left eigenvector of $\mathbf{M}_{PSR}$, which induces a relative trustworthy score vector, $\mathbf{r}_d$, including positive and negative values.
- Personalized Negative Rank (PNR) [3]: PNR is a heuristic method computing $\mathbf{r}^-$ as follows:

$$\text{PNR}(\mathbf{r}^-) = \text{RWR}(\mathbf{r}^+) - \text{PSR}(\mathbf{r}_d)$$

- Troll-Trust Model (TR-TR) [7]: TR-TR is a variant of PageRank where trustworthiness of users and edge weights are modeled as probabilities. TR-TR computes a trust score vector by propagating trustworthiness of users multiplied by outgoing edges' probabilistic weights which represent reliability between users. The final result of TR-TR corresponds to a relative trust score vector in terms of a default trustworthy β (e.g., $\beta = 0.5$). We can also calculate a personalized ranking for TR-TR by setting a seed node's trustworthiness to 1 per each iteration.

[1]SALSA [4] is a normalized version of HITS [2].

II. Description of Datasets

- LiveJournal: LiveJournal is a on-line community with almost 10 million members. Members in LiveJournal maintain journals and blogs, and connect other members who they think as friends. LiveJournal is a directed unsigned graph containing $4,847,571$ nodes and $68,475,391$ edges.
- Epinions: Epinions is an online review website where users show their views toward each other with positive and negative signs. Epinions is regarded as a directed graph comprising $131,828$ nodes and $841,372$ edges whose about 85.3% are positive.
- Slashdot: Slashdot is a technology-related news website, in which users can rate each other positively or negatively corresponding to friends and foes. Thus, Slashdot is a directed signed network containing $79,120$ nodes and $515,397$ edges whose about 76.1% are positive.
- Wikipedia: Wikipedia is a free encyclopedia that can be created and modified by users around the world. Wikipedia is maintained by some administrators who has additional authority to delete copyright violation, block malicious users and so on. These administrators are elected via a public discussion or a vote. Users can vote positively, negatively or neutrally on their candidates. This dataset contains $7,118$ nodes and $103,675$ edges whose about 78.4% are positive.

III. Description of Evaluation Measurements

A. Sign Prediction

- Accuracy: Accuracy is the proportion of successful results among the total number of cases.

B. Troll Identification

- Mean Average Precision (MAP@k): MAP@k is the mean of average precisions, AP@k, for multiple queries. Suppose that there l trolls to be identified. Then, AP@k is defined as follows:

$$\text{AP}@k = \frac{1}{min(l,k)}\left(\sum_{t\in\mathbf{T}} \text{Precision}@t\right)$$

  where Precision@t is the precision at the cut-off t. Note that $\mathbf{T} = \{t | \mathbb{I}_{\text{REL}}(\mathcal{R}[t]) = 1 \text{ for } 1 \le t \le k\}$ where $\mathcal{R}[t]$

denotes the index of the user that the ranking $\mathcal{R}$ returns at position t, and $\mathbb{I}_{\text{REL}}(\mathcal{R}[t])$ is 1 for those t in the ranking that contain a troll [1]. Hence, for N queries, MAP@k is defined as follows:

$$\text{MAP@}k = \frac{1}{N}\left(\sum_{i=1}^{N} AP@k\right)$$

- Normalized Discount Cumulative Gain (NDCG@k): NDCG is the normalized value of Discount Cumulative Gain (DCG), which is defined as follows:

$$\text{DCG@}k = rel_1 + \sum_{i=2}^{k} \frac{rel_i}{log_2(i)}$$

where rel_i is the user-graded relevance score for the i-th ranked item. Then, NDCG@k is obtained by normalizing using Ideal DCG(IDCG), which is the DCG for the ideal order of ranking, as follows:

$$\text{NDCG@}k = \frac{\text{DCG@}k}{\text{IDCG@}k}$$

To measure NDCG, we need user-graded relevance scores, but there are no such scores for the troll list. Hence, in a ranking, we set 1 for a troll user, and 0 for a normal user as relevance score.

- Precision@k and Recall@k: Precision@k (Recall@k) is the precision (recall) at the cut-off k in a ranking result. Precision@k is the ratio of identified trolls in the top-k ranking, and Recall@k is the ratio of identified trolls in the total trolls.
- Mean Reciprocal Rank (MRR@k): MRR@k is the mean of the reciprocal rank (RR) for each the top-k query response. RR is the multiplicative inverse of the rank of the first correct answer. Hence, for N multiple queries, MRR@k is defined as follows:

$$\text{MRR@}k = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{rank_i}$$

where $rank_i$ the rank position of the first relevant item in the top-k ranking. If there is no relevant item in the ranking for the i-th query, the inverse of the rank, ${rank_i}^{-1}$, becomes zero.

Precision@k and Recall@k are useful when we want to know how many relevant items are searched by a ranking method. MAP@k and NDCG@k are useful when a ranking order is important as well as relevant items. If relevan items are highly ranked, then those values are high. MRR@k is useful when we want to know how quickly a relevant item is appeared to a querying user.

References

[1] J. Kalofolias. Towards mean average precision. https://www.lsv.uni-saarland.de/fileadmin/teaching/snlp/ss15/MAP-AP.pdf.

[2] J. M. Kleinberg. Hubs, authorities, and communities. *ACM Computing Surveys (CSUR)*, 31(4es):5, 1999.

[3] J. Kunegis, A. Lommatzsch, and C. Bauckhage. The slashdot zoo: mining a social network with negative edges. In *Proceedings of the 18th international conference on World wide web*, pages 741–750. ACM, 2009.

[4] R. Lempel and S. Moran. Salsa: the stochastic approach for link-structure analysis. *ACM Transactions on Information Systems (TOIS)*, 19(2):131–160, 2001.

[5] A. Y. Ng, A. X. Zheng, and M. I. Jordan. Stable algorithms for link analysis. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 258–266. ACM, 2001.

[6] M. Shahriari and M. Jalili. Ranking nodes in signed social networks. *Social Network Analysis and Mining*, 4(1):1–12, 2014.

[7] Z. Wu, C. C. Aggarwal, and J. Sun. The troll-trust model for ranking in signed networks. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, pages 447–456. ACM, 2016.