Quantifying and Reducing Imbalance in Networks

Yoosof Mashayekhi[1], Bo Kang[1], Jefrey Lijffijt[1], and Tijl De Bie[1]

Ghent University, Ghent, Belgium
{yoosof.mashayekhi,bo.kang,jefrey.lijffijt,tijl.debie}@ugent.be

Abstract. Real-world data can often be represented as a heterogeneous network relating nodes of different types. For example, envision a job market network, where nodes may be job seekers, skills, and jobs, and where links to skills could indicate having that skill (if linked to a job seeker) or having the skill as a requirement (for jobs). It can be relevant to consider the *imbalance* in such a network between the nodes of different types. In the example, this imbalance could correspond to the mismatch between supply and demand of jobs due to a mismatch in skills. Identifying and reducing such imbalance is a problem of great significance.
We introduce a quantification of the imbalance in a network between two sets of nodes (nodes of different types, attributes, etc.) based on the embedding of a network, i.e., a real-valued vector space representation of the network nodes. Moreover, we introduce an algorithm named `GraB` (Graph Balancing) which ranks unconnected node pairs according to how well they would reduce the imbalance in a network if an edge were added between them. E.g., in the job network, `GraB` could be used to rank skills that job seekers do not yet have but could strive to acquire, to move them closer in the embedding towards an area where there is an abundance of jobs, and hence to reduce job market imbalance. We evaluated `GraB` on several datasets, including a job market network, and find that `GraB` outperforms baselines in reducing network imbalance.

Keywords: Network Imbalance · Network Embedding · Representation learning.

1 Introduction

Motivation: In some networks, an undesirable imbalance between two sets of nodes in a network exists. Let us consider a job market network, where there are three sets of nodes: job vacancies, job seekers, and skills, and where job vacancies are connected to the skills they require, and job seekers are connected to the skills they have and possibly to the job vacancies they apply for.

Imagine there are many Python developers seeking a job, and few vacancies requiring Python programming, while there are many vacancies requiring Java programming but few Java developers. As a result, Java jobs would remain unfilled and many Python programmers would remain unemployed. With an ever

faster evolving job market, such imbalances are increasingly common and serious, harming job market efficiency and ultimately the economy. Thus, quantifying such imbalances would provide policy makers with an objective picture of the current state of affairs.

Moreover, the ability to quantify imbalance also opens up the possibility of trying to reduce it through targeted interventions and incentives. While policy makers may not be able to influence employers to shift their requirements, they can provide courses and training material for specific worker profiles lacking sought-after skills, to shift their area of expertise and better meet the demand of the job market. In network terms, is equivalent to adding a certain number of links connecting job seekers (let us call this the set of *source nodes*) to skills (*auxiliary nodes*), to reduce the imbalance between job seekers (*source nodes*) and job vacancies (*target nodes*).

Our approach: Given an undirected network by $G = (V, E)$, where V and E are the sets of nodes and links respectively. Moreover, we define three sets of nodes, namely source nodes S, target nodes T, and auxiliary nodes U. Sets S, T, and U are three disjoint subsets of V.

Given such a network with sets S and T, we define its *imbalance* as the cost of a so-called optimal fractional perfect b-matching [1] between these sets in a complete bipartite network constructed using the two sets of nodes. Fully specifying the fractional perfect b-matching problem requires the definition of the *cost* of matching any pair of nodes from S and T. As the (inverse of the) distance between a pair of nodes' embeddings in a network embedding usually represents some type of affinity between these nodes, we propose to let this cost be based on this distance.

Next, we define the problem of adding a limited number of links between sets S and U, to *reduce the imbalance* between sets S and T. Adding links will change the embedding of the network and thus modify the cost of matching node pairs. We propose a method called Graph Balancing (`GraB`) to tackle this problem, which is based on a local approximation of the imbalance that we may compute analytically, thus providing the necessary scalability.

Example: To understand the relevance of the embedding, consider the sample network shown in Figure 1a. The figure shows three clusters of nodes. In the top-left cluster, source and target nodes are mixed, while the bottom-left cluster only contains nodes in S while the top-right cluster only contains nodes in T. Our goal is to quantify which links between S and U (the green nodes here) would reduce the imbalance between S and T. Figure 1b shows the network embedding after adding the top 200 links from `GraB` to the network. Now, S and T are well-mixed in the network.

The **main contributions** of this paper are:

- In Section 2, we *define the imbalance* between two sets of nodes S and T, in a network and propose the *measure* $\boldsymbol{\psi}(\boldsymbol{D}, S, T)$ *for quantifying it*, where we compute the cost of matching node pairs using the embedding distances $\boldsymbol{D}$. We also formulate the novel problem of *reducing the imbalance* in a network by adding a limited number of links.

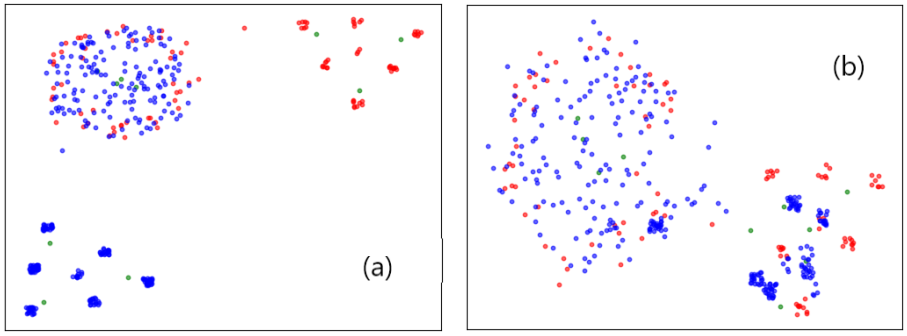

Fig. 1: A sample network. Blue, red and green nodes denote source, target and auxiliary nodes, respectively. (a) A 2D embedding of the original network. (b) A 2D embedding of the network after adding 200 links between source and auxiliary nodes using `GraB`, to reduce the imbalance between source and target nodes.

- In Section 3, we propose a novel generic method, *Graph Balancing (`GraB`)*, to optimally select a limited number of links to add to the network to reduce the imbalance. Because this is computationally challenging, we introduce a link utility that uses a local imbalance measure as a proxy and employ a greedy algorithm to select the links.
- In Section 4, we propose two *baselines* (a naive and a more intelligent baseline) for the novel problem of reducing the imbalance in a network. We perform an experiment on several datasets to compare the performance of `GraB` and baselines in reducing the imbalance in a network embedding. The experiment shows that `GraB` outperforms the baselines in this task.
- In Section 5, we discuss the related work.

2 Network Imbalance

To generically define our proposed notion of imbalance, we develop it first for the concrete example of the job market. There, we define the imbalance as the cost of matching all job seekers to jobs. Here, we allow a job seeker and a job to be matched even if they are not connected by a link. Indeed, a link between a job seeker and a job vacancy might mean that the job seeker has applied for the job or has otherwise expressed an interest, not necessarily that they were employed for that job. Moreover, the absence of a link does not imply that the job seeker would not be a good candidate for the job. *It is this property that distinguishes our work from the literature on combinatorial matching problems in graphs.*

However, the skills to which the job seeker and job vacancy are linked, and jobs vacancies they *are* linked to, provide information on whether the job seeker is suited for the job; and the more suited, the smaller the cost of a match between them should be. Hence, the cost of matching a job seeker and a job should be a function of the network structure, and adding or removing skills to that job

seeker or job should influence the cost of matching them. Hence, the cost could be defined using any model that provides link costs or link probabilities (so the cost of node pairs could be computed based on them) based on the network structure. In this paper, we investigate using the embedding of the network for this, as it is a state-of-the-art approach to summarize the network structure, where proximity between node embeddings reflects the probability that both nodes *should be* linked. Formally, we define the imbalance in a network as follows:

Definition 1 (Imbalance). *Consider a network $G = (V, E)$, two disjoint non-trivial subsets of its nodes referred to as the source nodes $\emptyset \subset S \subset V$ and the target nodes $\emptyset \subset T \subset V$ with $S \cap T = \emptyset$, and the costs d_{ij} of matching the pairs of nodes i and j for all $i \in S$ and $j \in T$, arranged in a matrix $\boldsymbol{D} = [d_{ij}]$. Moreover, consider the complete edge-weighted and node-weighted bipartite network $G' = (V' = S \cup T, E' = S \times T)$, with weight of edge $\{i, j\}$ with $i \in S$ and $j \in T$ equal to d_{ij}, and weight of node $i \in S$ equal to $w_i = \frac{1}{|S|}$ and weight of node $j \in T$ equal to $w_j = \frac{1}{|T|}$[1]. Then the imbalance between nodes in sets S and T, denoted as $\boldsymbol{\psi}(\boldsymbol{D}, S, T)$, is defined as the cost of the minimum cost fractional perfect b-matching in G'.*

Computing the imbalance defined in this way can be done by solving a linear programming problem for finding the matching $\boldsymbol{F} = [f_{ij}]$ in G' that minimizes the overall cost:

$$C = \sum_{i \in S} \sum_{j \in T} f_{ij} d_{ij},$$
$$\text{s.t.} \quad f_{ij} \geq 0 \quad \forall (i,j) \in S \times T, \quad \sum_{j \in T} f_{ij} = w_i \quad \forall i \in S, \quad \sum_{i \in S} f_{ij} = w_j \quad \forall j \in T.$$

The imbalance measure ψ is defined as the minimum cost of the optimal matching. (The matching itself is not of interest to us for the purposes of this paper.). As discussed before, our approach is to use the distance of nodes in the embedding space as the matching cost d_{ij} of each node pair.

As the cost of matching node pairs depends on the structure of the network, it will change by modifying the network. Motivated by the job market example, we propose the operation of adding links between source (job seekers) and auxiliary (skills) nodes as the kind of modification that can be made. More formally, we introduce the following problem:

Problem 1 (Imbalance Reduction). Given a network $G = (V, E)$ and three mutually disjoint sets of nodes *source nodes* $\emptyset \subset S \subset V$, *target nodes* $\emptyset \subset T \subset V$ and *auxiliary nodes* $\emptyset \subset U \subset V$, the cost of matching each pair of nodes $\boldsymbol{D} = [d_{ij}]$ (that depends on the network structure G), and imbalance measure $\psi(\boldsymbol{D}, S, T)$, find the optimal k links $\mathcal{E}$ connecting nodes from set S with nodes from set U, that reduce the imbalance between the nodes in sets S and T.

[1] The total weight of nodes in S and T would be equal to 1.

3 Reducing Network Imbalance: GraB

The exact minimization problem of adding k links to a network in order to maximally reduce the imbalance, as defined in Definition 1 amounts to finding a set of links that jointly minimize the imbalance. This exact approach may be computationally intractable when the number of candidate links is large, since it may require computing the imbalance reduction for every possible set of k links. Moreover, to compute the reduction in imbalance we need to re-embed the network and compute the imbalance again. Recomputing the embedding is computationally demanding, and is practically impossible to do. Motivated by these two problems, our approach is as follows.

To overcome the combinatorial problem of considering every possible set of k links, we employ a greedy algorithm to select the top k links with the highest benefits. Instead of recomputing the embedding and imbalance after adding each link to calculate its benefit, we introduce a link utility based on the fact that moving source nodes from areas with a lot of source nodes and few target nodes (which means that only a few of the source nodes could be matched with the closeby target nodes and the rest should be matched with target nodes at more distance) to areas with fewer source nodes and more target nodes would reduce the matching cost (imbalance). We compute the discrepancy between source and target nodes in an area in the embedding space using a local imbalance measure, which we calculate using the kernel density estimation [11] of source and target nodes in the embedding space. Next, we compute the link utility by calculating how the embedding of a node would change after adding a link to it, and in which direction the local imbalance measure improves. The pseudo-code of GraB is presented in Appendix A

4 Experimental Evaluation

We evaluate the performance of GraB by conducting several experiments on three datasets: VDAB [2] (a job market network), Weibo [14] (a social network) and Movielens [6] (a movie recommender system). More details of networks are provided in Appendix B. We compare GraB with S-GraB (a simpler version of GraB), ROV [4] (a method to reduce controversy in a network), SSW [10] (a method to reduce the average shortest path in a network), S-Random (simple random method) and I-Random (a random method which employs the greedy algorithm in GraB). In the quantitative experiments, we compare methods in terms of ψ (Definition 1). We conduct the experiments on CNE [8] with dimensions 2 and 4. The source code for repeating the experiments and the preprocessed public datasets are available[3].

Table 1 shows the result of our experiment. GraB reduces ψ over the main graph and outperforms the other methods in all datasets. S-GraB also outperforms other baselines in most datasets.

[2] https://www.vdab.be/

[3] https://tinyurl.com/18qbxirs

Table 1: ψ after adding 100 links to each network. Best performance per dataset is highlighted in bold.

Dataset	Main Graph	S-Random	I-Random	SSW	ROV	S-GraB	GraB
VDAB-CNE2	0.2357	0.2335	0.2286	0.2351	0.2362	0.2254	**0.2245**
Weibo-mf-CNE2	0.5729	0.5650	0.5479	0.5361	0.56891	0.5191	**0.4955**
Weibo-fm-CNE2	0.5724	0.5750	0.5419	0.6763	0.5739	0.4917	**0.4425**
Movielens-CNE2	0.4667	0.4653	0.4654	0.4501	0.4652	0.4458	**0.4400**
VDAB-CNE4	0.4211	0.4206	0.4164	0.4195	0.4213	0.4170	**0.4137**
Weibo-mf-CNE4	0.6432	0.6423	0.6245	0.6252	0.6506	0.6269	**0.6049**
Weibo-fm-CNE4	0.6536	0.6486	0.6290	0.7175	0.6446	0.6235	**0.5895**
Movielens-CNE4	0.6031	0.5927	0.5990	0.5856	0.6002	0.5748	**0.5686**

5 Related Work

Imbalance in the workforce has been studied in various systems [15,13]. They are mostly domain-specific and they analyze the supply and demand based on domain-specific features such as educational training program length, retirement, and salary. Previous studies in this area also lack a global measure to quantify the imbalance between two different entities. In contrast to this, we tackle the problem from a graph analysis approach. We propose a quantification of the imbalance in the network using its embedding and a method to reduce the imbalance by adding links to the network.

Another line of research that focuses on matching problems [9,12]. Finding the cost of the fractional perfect b-matching [1] with minimum cost in bipartite networks is related to our work. Our work differs from the studies focusing on matching, since we do not address the computational problem of how to find the matching, we only use the cost of the matching to quantify the imbalance. Moreover, we add links to the network (not directly between the two sets of nodes of interest), to change the cost of links between the two sets of nodes, and hence, reduce the imbalance in the network.

There are also several papers aiming to add links to a network to modify the network structure and make it more cohesive, where cohesiveness is quantified using network properties such as shortest paths [10], diameter [3,2], information unfairness [7], controversy [4] and structural bias [5]. Our approach to modify the graph and the objective function is different from the mentioned methods.

Acknowledgements The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013) / ERC Grant Agreement no. 615517, from the Flemish Government under the "Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen" programme, and from the FWO (project no. G091017N, G0F9816N, 3G042220). Part of the experiments were conducted on pseudonimized HR data generously provided by VDAB.

References

1. Behrend, R.E.: Fractional perfect b-matching polytopes i: General theory. Linear Algebra and its Applications **439**(12), 3822–3858 (2013)
2. Bilò, D., Gualà, L., Proietti, G.: Improved approximability and non-approximability results for graph diameter decreasing problems. Theoretical Computer Science **417**, 12–22 (2012)
3. Demaine, E.D., Zadimoghaddam, M.: Minimizing the diameter of a network using shortcut edges. In: Scandinavian Workshop on Algorithm Theory. pp. 420–431. Springer (2010)
4. Garimella, K., De Francisci Morales, G., Gionis, A., Mathioudakis, M.: Reducing controversy by connecting opposing views. In: Proceedings of the Tenth ACM International Conference on Web Search and Data Mining. pp. 81–90 (2017)
5. Haddadan, S., Menghini, C., Riondato, M., Upfal, E.: Repbublik: Reducing the polarized bubble radius with link insertions. arXiv preprint arXiv:2101.04751 (2021)
6. Harper, F.M., Konstan, J.A.: The movielens datasets: History and context. Acm transactions on interactive intelligent systems (tiis) **5**(4), 1–19 (2015)
7. Jalali, Z.S., Wang, W., Kim, M., Raghavan, H., Soundarajan, S.: On the information unfairness of social networks. In: Proceedings of the 2020 SIAM International Conference on Data Mining. pp. 613–521. SIAM (2020)
8. Kang, B., Lijffijt, J., De Bie, T.: Conditional network embeddings. arXiv preprint arXiv:1805.07544 (2018)
9. Kolmogorov, V.: Blossom v: a new implementation of a minimum cost perfect matching algorithm. Mathematical Programming Computation **1**(1), 43–67 (2009)
10. Parotsidis, N., Pitoura, E., Tsaparas, P.: Selecting shortcuts for a smaller world. In: Proceedings of the 2015 SIAM International Conference on Data Mining. pp. 28–36. SIAM (2015)
11. Parzen, E.: On estimation of a probability density function and mode. The annals of mathematical statistics **33**(3), 1065–1076 (1962)
12. Ramshaw, L., Tarjan, R.E.: On minimum-cost assignments in unbalanced bipartite graphs. HP Labs, Palo Alto, CA, USA, Tech. Rep. HPL-2012-40R1 (2012)
13. Willis, G., Woodward, A., Cave, S.: Robust workforce planning for the english medical workforce. In: Conference Proceedings, The 31st International Conference of the System Dynamics Society (2013)
14. Zhang, J., Liu, B., Tang, J., Chen, T., Li, J.: Social influence locality for modeling retweeting behaviors. In: Twenty-Third International Joint Conference on Artificial Intelligence (2013)
15. Zurn, P., Dal Poz, M.R., Stilwell, B., Adams, O.: Imbalance in the health workforce. Human resources for health **2**(1), 1–12 (2004)

Appendix A

Algorithm 1 shows the greedy link selection in `GraB`. Algorithm 2 the complete generic method `GraB` to select k links to add to the network.

Appendix B

The datasets used in evaluation are described below:

Algorithm 1: Greedy Link Selection

Input: NE(network embedding method), $\boldsymbol{A}$ (adjacency matrix), S (source nodes), T (target nodes), U (auxiliary nodes), *length* (number of links to select)
Output: *top_links*
Function `SelectCandidateLinks`(NE, $\boldsymbol{A}$, S, T, U, *length*):
    $\boldsymbol{X} = NE$.Embeddings($\boldsymbol{A}$)
    $P = NE$.LinkPrediction($\boldsymbol{X}$, $\boldsymbol{A}$) `// Find link probabilities from network embedding method`
    $\boldsymbol{B} = [0]_{n \times n}$ `// Benefit matrix`
    foreach i *in* S **do**
        foreach j *in* U **do**
            if a_{ij}==*0* **then**
                $\boldsymbol{B}_{ij} = \frac{\partial \delta}{\partial a_{ij}}(\boldsymbol{x}_i, \boldsymbol{X}, S, T)$
            end
        end
    end
    top_links = []
    for *idx* $\leftarrow 1$ **to** *length* **do**
        $(i, j) = \text{Max}(\boldsymbol{B})$ `// Selects the link with the maximum benefit`
        top_links[idx] = (i, j)
        $\boldsymbol{B}_i = 0$ `// Setting row` i `of matrix` B `to zero, because one link per source node is allowed.`
    end
    return *top_links*

Algorithm 2: Graph Balancing (`GraB`)

Input: NE(network embedding method), $\boldsymbol{A}$ (adjacency matrix), S (source nodes), T (target nodes), U (auxiliary nodes), k (number of links to add), b (batch size), l (batch size coefficient)
Output: *links* (k links)
Function `GraB`(NE, $\boldsymbol{A}$, S, T, U, k, b, l)**:**
    $links = []$, $links\_idx = 1$
    $\boldsymbol{X} = NE$.Embeddings($\boldsymbol{A}$)
    while $links\_idx \leq k$ **do**
        $candidate\_list$ = SelectCandidateLinks(NE, $\boldsymbol{A}$, S, T, U, $b \cdot l$)
        $\boldsymbol{X}\_new$, $\boldsymbol{A}\_new$ = NE.ReEmbed($\boldsymbol{X}$, $\boldsymbol{A}$, $candidate\_list$)
         `// Re-embed the network after adding the` $candidate\_list$ `links`
        $current\_batch\_list = []$, $idx = 1$
        foreach *(i, j) in candidate_list* **do**
            if $\delta(\boldsymbol{x}_i, \boldsymbol{X}, S, T) < \delta(\boldsymbol{x}\_new_i, \boldsymbol{X}\_new, S, T)$ **then**
                $links[links\_idx] = (i, j)$
                $links\_idx$ += 1
                $current\_batch\_list[idx] = (i, j)$
                idx += 1
                if $idx > b$ **then**
                    break
                end
            end
        end
        $\boldsymbol{X}$, $\boldsymbol{A}$ = NE.ReEmbed($\boldsymbol{X}$, $\boldsymbol{A}$, $current\_batch\_list$) `// Re-embed the network after adding the` $current\_batch\_list$ `links`
    end
    return *links*

VDAB [4]: VDAB is the employment service of Flanders in Belgium. It provides a platform for job seekers to find jobs. The dataset contains a sample of the applications made by job seekers to available job vacancies from January 2018 until October 2020. We construct the job market network with three sets of nodes: job seekers, job vacancies, and skills. Job vacancies are connected to job seekers that have applied for them and to the skills they require. Our goal is to reduce the imbalance between job seekers (source nodes) and job vacancies (target nodes) by adding links connecting job seekers with skills. This could be seen as teaching a group of job seekers some skills, in a way that balances the job market network.

Weibo [14]: Weibo is the most popular Chinese microblogging service. The dataset contains tweets of the users and the topic distribution of each tweet. We construct the network with three sets of nodes: male users, female users, and topics. Users are connected to their friends (reciprocal follow relationships) and the top topics of their tweets. To find the top topics for each tweet, we first sort the topic probabilities (relevance of topic for the tweet) in descending order. Next, we select the top topics until the difference of the probabilities of two consecutive topics is greater than a very small threshold (1e-6). We select a sample from the dataset by only considering tweets for the first year of the data. Our goal is to reduce the imbalance between males (source nodes or target nodes) and females (target nodes or source nodes) by adding new links connecting males/females with topics (auxiliary nodes). It is more like recommending tweets with specific topics to the users to increase their interest in that topic. In the experiments, we call this dataset **Weibo-mf** if males are the source nodes and females are the target nodes. Otherwise, we call the dataset **Weibo-fm**.

Movielens [6]: MovieLens is a web-based movie recommender system. The dataset contains 100000 user ratings on movies. We construct the network with three sets of nodes: users, movies, and movie genres. There is a link between each user and movie for each rating. We also connect each movie to its genres. Our goal is to reduce the imbalance between movies (source nodes or target nodes) and users (target nodes or source nodes) by adding new links connecting movies with genres (auxiliary nodes).

Table 2 shows the main statistics of each of the networks.

Table 2: Main statistics of the networks used for evaluation.

Dataset	VDAB	Weibo-mf	Weibo-fm	Movielens
Nodes	5016	1364	1364	2644
Source Nodes	2358	822	442	1682
Target Nodes	2463	442	822	943
Auxiliary Nodes	195	100	100	19
Links	48451	14308	14308	102893
Average Degree	19	20	20	77

[4] `https://www.vdab.be/`