Machine Learning Methods with Link-Level Features

Koffka Khan

Department of Computing and Information Technology, The University of the West Indies, St. Augustine Campus, TRINIDAD AND TOBAGO

Abstract

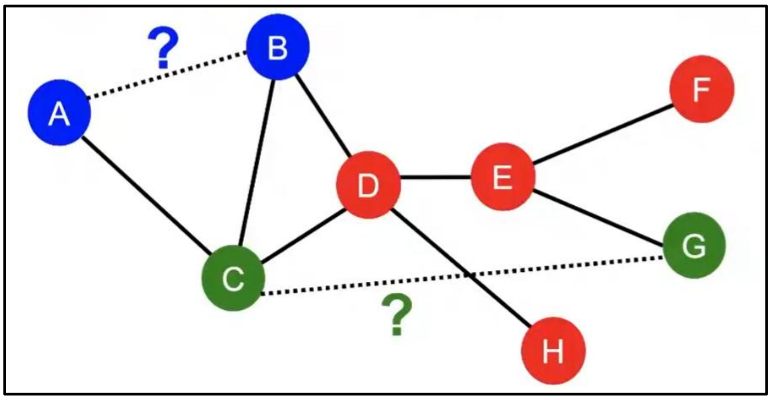
This paper describes graph link level features. We describe three different kinds of them. We discuss distance-based characteristics that consumers might use, such as the shortest path between two nodes, which does not account for neighborhood overlaps. Then we discuss neighborhood overlap metrics like Common neighbors, Jaccard coefficient, and the Adamic-Adar index that measures how manyneighbors a pair of nodes share in a fine-grained manner. The issue with this is that the metric will return a value of 0 for nodes that are more than two hops away or that do not have any neighbors. The global neighborhood overlap metrics, for instance, like Katz use the global graph structure to give us a score for a pair of nodes, and the Katz index counts the number of pets of all lands between a pair of nodes where these paths are discounted exponentially with their length.

Keywords: graph, link, features, common neighbors, Jaccard coefficient, Adamic-Adar index, Katz index

1. Introduction

We keep looking into classic machine learning methods as they relate to graph level predictions. And now that we have them, our attention will be on link prediction jobs and features that capture the structure of linkages in a specific network [15]. Therefore, the following is the link level prediction task [16]. Predicting new links (see Figure 1) based on the network's current links is the task at hand. This means that, at test time, we must evaluate all node pairs that are not yet connected, rank them, and then declare that, as predicted by our method, the top k node pairings are the connections that will form in the network.

Figure 1: Link prediction.



Designing features for a pair of nodes is crucial. Let's concatenate the characteristics of node A, features of the node B, and train a model on that type of representation, as can be done at the node level tasks [7]. That, however, would be quite unsatisfying because frequently this would lose a lot of crucial information regarding the connection between the two nodes in the network.

We will therefore approach this link prediction task in a two-way manner. We have two options for formulation. Simply stating that links in the network are, let's say, randomly missing is one way to put it. Then, given a network, we will randomly remove a certain amount of links and attempt to anticipate back those linkages using our machine learning technique. That is a particular kind of formulation.

The second kind of formulation is that we will forecast linkages over time [19]. Accordingly, if we have a network that naturally changes over time, such as a citation network [1], social network [8], or collaboration network [20], we can say, "We are going to look at a graph between time zero and time zero-prime, and based on the edges and the structure up to this time t-0-prime we are going to then output a ranked list L of links that we predict will occur in the future." Let's assume that will manifest between t1 and t1-prime.

We may evaluate this type of method by ranking the prospective edges produced by our algorithm and comparing them to the edges that in fact did really occur in the future. This is because we know that n new links will appear in the future. While the links missing at random type formulation, for instance, is more beneficial, this type of formulation is useful or natural for networks that expand over time, such as transaction networks and social networks, where edges are always added.

For instance, for static networks like protein-protein interaction networks [11], where we can assume, despite the fact that this assumption is actually heavily violated, that, you know, biologists are testing kind of haphazard connections between proteins and we'd like to infer what other connections in the future or for biologists to discover in the future, or which links should they probe with their lab experiments. Of course, in practice biologists don't just randomly explore the physical network of protein-protein interactions. They are, you know, greatly impacted by one other's successes. In essence, some areas of this network are now significantly underexplored while others are suddenly widely studied.

Now that we have these two formulations, let's start considering how we will supply a feature description for a particular pair of nodes. The plan is to compute a score c(x, y) for a pair of nodes called x, y. A score, for instance, can represent the quantity of nodes x and y that share neighbors. Then, after sorting all x, y pairings according to decreasing c scores, we'll forecast that the top end pairs will be the new links that will enter the network.Following the test period's conclusion, we can compare these two lists, watch which links actually appear, and assess how effectively our strategy and algorithm are performing.

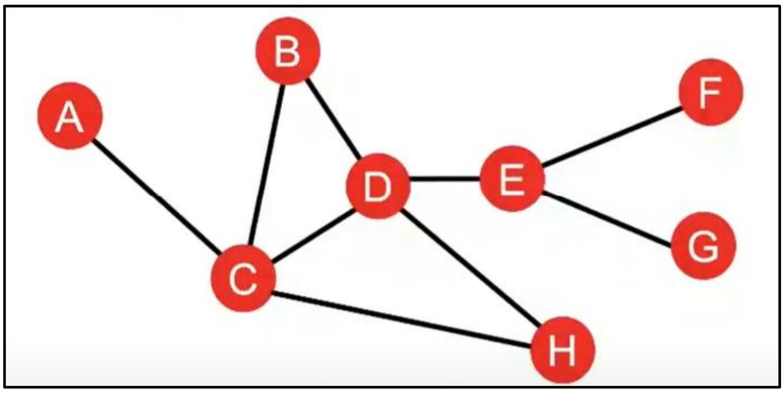
We'll go through three distinct approaches to structuring or developing a descriptor of the connection between two network nodes. Distance-based features, local neighborhood overlap features, and global neighbor overlap features will all be covered. And the objective is to characterize the relationship between a particular pair of nodes so that, from this relationship, we can then infer or determine if there is or isn't a link between them.

This paper consists of six sections. In Section 2 we discuss Distance-based features, while in Section 3 we discuss Local Neighborhood Overlap features. Global Neighborhood Overlap featuresis illustrated in Section 4. The conclusion and references are given in Sections 5 and 6, respectively.

2. Distance-based features

Thus, we begin by discussing the distance-based feature [4]. This is very natural. This is how we can conceptualize and describe the shortest path distance between two nodes. So, for instance, if nodes B and H are present, the shortest path length between them is two.

Figure 2: Graph: shortest-path between two vertices/nodes.



Therefore, this feature would have a value of two. However, if you look at the figure, you'll see that it does not measure what this metric does not capture; it measures distance but not how much a neighborhood overlaps or how strong a relationship there is. Since nodes B and H, for instance, may be found in this network, they truly have two friends in common.

3. Local NeighborhoodOverlap features

As a result, there is a stronger connection between them in this situation (continuing our discussion from Section 2). The connection between nodes D and F, for instance, is not as strong because there is only one path there, compared to the two paths present [D-E-F]. So, asking two nodes, "Okay, how many neighbors do you have in common?" would be a good method to try to measure the strength of their link. How many nodes have friends in common with one another? The idea of local neighborhood overlap [10], which measures the number of neighbors that two nodes, v1 and v2, share, captures this.Saying "what is the number of common neighbors" is one way to express this, isn't it? We take the intersection of the neighbors of nodes, see Equation 1 showing common neighbors between two nodes v_1 and v_2.

$$|N(v_1) \cap N(v_2)| \qquad (1)$$

The Jaccard coefficient [14], which takes the intersection's size and divides it by the union's size, is a normalized version of this same concept between two nodes v_1 and v_2, see Equation 2. The problem with common neighbors is that higher degree nodes are obviously more likely to have neighbors with other nodes. While stating what is the union of the number of neighbors of the two nodes, we are in a sense trying to normalize the Jaccard coefficient to some degree.

$$\frac{|N(v_1) \cap N(v_2)|}{|N(v_1) \cup N(v_2)|}$$ (2)

Additionally, the Adamic-Adar index [18] is another type of local neighborhood overlap metric that actually performs fairly well in practice, see Equation 3. Simply put, this is indicating that we should total all of the neighbors that nodes v_1 and v_2 share and then take the neighbor with the highest degree.

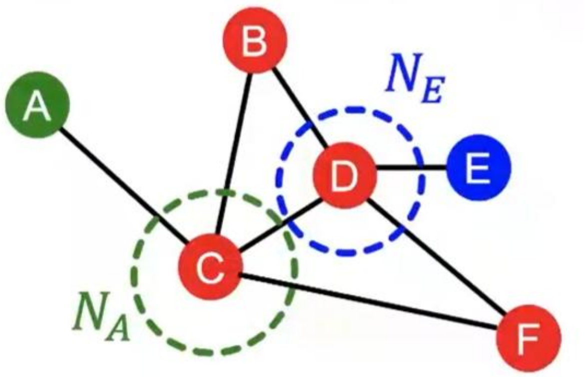
$$\sum_{u \in N(v_1) \cap N(v_2)} \frac{1}{\log(k_u)}$$ 3)

Basically, the concept is to measure how many neighbors a pair of nodes share, although the significance of an unbalanced neighbor is modest and declines with these degrees.

Therefore, having a large group of low degree neighbors in common is preferable to having a large group of well-known, well-connected superstars. This is a feature of a social network that functions incredibly well. Of course, the issue with local network neighborhood overlap is that it is constrained by the fact that it always returns 0 if two nodes do not share any neighbors.

Because they don't share any neighbors, for instance, nodes A and E are more than two hops apart in this scenario if we wanted to determine the neighborhood overlap between them, see Figure 3. The value of that it will be returned to will then, if only in such instances, always be zero. In fact, though, there's still a chance that these two nodes will eventually link. We then develop a global neighborhood overlap matrix to address this issue. This is the whole restriction of just considering two-hop distances and pathways between pairs of nodes, as opposed to taking into account all other distances or the entire graph.

Figure 3: Graph: Local Neighborhood Overlap.



4. GlobalNeighborhoodOverlap features

Therefore, let's examine global neighborhood overlap measurements [21] right now (continuing our discussion from Section 3). The statistic we'll discuss is known as the Katz index [12], and it counts the number of paths—of various lengths—that can be taken between a particular pair of nodes. We must now resolve two issues in this situation. The first question is: How can we determine the number of pathways of a certain length that connect two nodes? Actually, this can be calculated in a very elegant manner using the graph adjacency matrix's powers. So let me quickly illustrate or quickly demonstrate why this is true.

So, first things first, let me explain the powers of the adjacency matrix [6], is that right? The purpose of this demonstration is to demonstrate that calculating the number of pathways between two nodes may be done by computing the graph adjacency matrix, or by multiplying the graph adjacency matrix by itself. Recall that the first graph adjacency matrix has a value of 1 for each element u, v if and only if nodes u and v are connected, see Equation 4.

$$\boldsymbol{A}_{uv} = 1 \text{ if } u \in N(v) \quad (4)$$

The number of pathways of length K connecting nodes u and v is counted by the superscript capital K in the expression P_{uv}, see Equation 5.

$$\boldsymbol{P}_{\boldsymbol{uv}}^{(\boldsymbol{K})} = \#\text{paths of length } \boldsymbol{K} \text{ between } \boldsymbol{u} \text{ and } \boldsymbol{v} \quad (5)$$

Our objective is to demonstrate that if we want to know the number of pathways of length K, we must compute A to the power of K, and entry u, v will tell us how many paths there are, see Equation 6.

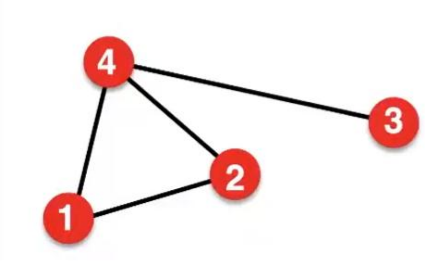
$$\boldsymbol{P}^{(\boldsymbol{K})} = \boldsymbol{A}^{\boldsymbol{k}} \quad (6)$$

The Kth power of A counts the number of pathways with a specific length since the capital K in this case is the same as a tiny case, see Equation 7.

$$\boldsymbol{P}_{\boldsymbol{uv}}^{(\boldsymbol{1})} = \#\text{paths of length 1 between } u \text{ and } v = \boldsymbol{A}_{\boldsymbol{uv}} \quad (7)$$

And if you think about it correctly, how many paths between two nodes of length one exist that are perfectly captured by the graph adjacency matrix? A value of 1 indicates that a pair of nodes are connected, and a value of 0 indicates that a pair of nodes are not connected.

Figure 4: Adjacency matrix powers.



We can now determine how many pathways [9] of length one there are between any two nodes. The question of how many pathways of length 2 there are between a pair of nodes u can now be asked. And we're going to accomplish this using a two-step process. We're going to accomplish this by splitting the path of length 2 into two paths, each of length 1. Determining the number of paths of length 1 that exist between each of u's neighbors and v is the notion, and we then add one to that number.

Figure 5: Graph: Global Neighborhood Overlap; symmetry $P^{1}_{12} = A_{12}$.

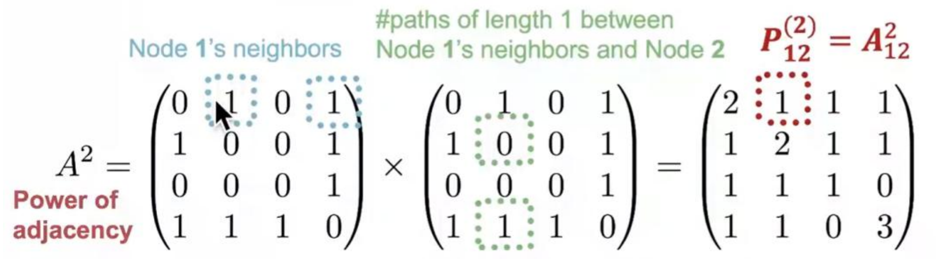
$$A = \begin{pmatrix} 0 & \boxed{1} & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

The number of pathways between nodes u and v of length 1 and length 2 is simply calculated by adding all of the nodes I that are neighbors of the starting node u together, multiplied by the number of paths that lead from this neighbor I to the target node v, see Equation 8. And now we will know how many pathways there are between u and v that are two lengths long. And as you can see, the adjacency matrix is being used as a replacement.

$$\boldsymbol{P^{(2)}_{uv} = \textstyle\sum_i A_{ui} * P^{(1)}_{iv} = \sum_i A_{ui} * A_{iv} = A^{2}_{uv}} \quad (8)$$

If you look at this, it is simply the adjacency matrix itself multiplied by the matrices that produced it, see Figure 6.

Figure 6: Power of adjacency.



$$P^{(2)}_{12} = A^{2}_{12}$$

$$A^2 = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix} \times \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 2 & 1 & 1 & 1 \\ 1 & 2 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 3 \end{pmatrix}$$

The adjacency matrix A squared now has this as its entry. Now that we have established that this is essentially the result of induction, we may keep doing this to obtain higher powers that count pathways with longer lengths as long as this is increasing. When we are interested in a given, let's say entry, these are entry, and these are Node 1's neighbors because A squared is A multiplied by itself.

These are the number of pathways between a node's neighbors and node number 2 that are of length 1. Thus, the number here will be 1, indicating that there is only one path of length 2 from Node 1 to Node 2. In order to account for pathways of length K between a pair of nodes in the network, powers of the adjacency matrix are used. As a result, we are now able to define the cuts index and have created the first component that will enable us to count and compute the cuts index since it enables us to count the number of paths there are for a given K between any two nodes.

However, we still have to decide how to accomplish this for all path lengths, ranging from one to infinite. The adjacency matrix itself tells us powers of length 1, squared gives us powers of, squared tells us pathways of length 2, and the adjacency matrix raised to the power l counts the number of paths of length l between a pair of nodes. As we previously stated, this is how we will compute the pets. The Katz index spans the range of one path length to infinity. Thus, the global neighborhood overlap in the Katz index [5] between nodes v_1 and v_2 is just the total of l from 1 to infinity.

$$S_{v_1 v_2} = \sum_{l=1}^{\infty} \boxed{\beta^l} \boxed{A^l_{v_1 v_2}} \quad \text{\#paths of length } l \text{ between } v_1 \text{ and } v_2 \qquad 0 < \beta < 1\text{: discount factor} \tag{9}$$

Equation 9 counts the number of pathways of length l between the nodes of v_1 and v_2, and Beta is raised to the power of l by a discount factor that devalues paths of larger lengths. What's intriguing about the Katz index right now is that we can compute this specific phrase in closed form, see Equation 10. This is really a closed form expression that will compute the sum with accuracy. This is also the reason why this is true or why there is inequality, depending on your perspective.

$$\boldsymbol{S} = \sum_{i=1}^{\infty} \beta^i \boldsymbol{A}^i = \underbrace{(\boldsymbol{I} - \beta \boldsymbol{A})^{-1}}_{\substack{= \sum_{i=0}^{\infty} \beta^i \boldsymbol{A}^i \\ \text{by geometric series of matrices}}} - \boldsymbol{I}, \tag{10}$$

We see that this is a straightforward geometric series for matrices [17], and to express it in closed form, all we need to do is take the identity matrix [2], subtract beta from it, multiply the adjacency matrix by that, flip it over, and then subtract the identity matrix once more.And for any pair of nodes, the entries in this matrix S will provide us with the Katz neighborhood overlap scores [3].

5. Conclusion

In conclusion, this paper describes graph link level features. We described three different kinds of them. We discussed distance-based characteristics that consumers might use, such as the shortest path between two nodes, which does not account for neighborhood overlaps. Then we discussed neighborhood overlap metrics like Common neighbors, Jaccard coefficient, and the Adamic-Adar index that measures how many neighbors a pair of nodes share in a fine-grained manner. The issue with this is that the metric will return a value of 0 for nodes that are more than two hops away or that do not have any neighbors. The global neighborhood overlap metrics, for instance, like Katz use the global graph structure to give us a score for a pair of nodes, and the Katz index counts the number of pets of all lands between a pair of nodes where these paths are discounted exponentially with their length.

6. References

1. Annarelli A, Battistella C, Nonino F, Parida V, Pessot E. Literature review on digitalization capabilities: Co-citation analysis of antecedents, conceptualization and consequences. Technological Forecasting and Social Change. 2021 May 1;166:120635.
2. Chen B, Peng D, Zhang J, Ren Y, Jin L. Complex Table Structure Recognition in the Wild Using Transformer and Identity Matrix-Based Augmentation. InInternational Conference on Frontiers in Handwriting Recognition 2022 (pp. 545-561). Springer, Cham.
3. Gao Z, Rezaeipanah A. A Novel Link Prediction Model in Multilayer Online Social Networks Using the Development of Katz Similarity Metric. Neural Processing Letters. 2022 Nov 24:1-23.
4. Garg S, Narayanam R, Bandyopadhyay S. A framework to preserve distance-based graph properties in network embedding. Social Network Analysis and Mining. 2022 Dec;12(1):1-3.

5. Ge J, Shi LL, Liu L, Shi H, Panneerselvam J. Edge intelligence- enabled dynamic overlapping community discovery and evolution prediction in social media data streams. Concurrency and Computation: Practice and Experience. 2021 Dec 23:e6786.
6. Ghorbani M, Li X, Zangi S, Amraei N. On the eigenvalue and energy of extended adjacency matrix. Applied Mathematics and Computation. 2021 May 15;397:125939.
7. Jin W, Derr T, Wang Y, Ma Y, Liu Z, Tang J. Node similarity preserving graph convolutional networks. InProceedings of the 14th ACM international conference on web search and data mining 2021 Mar 8 (pp. 148-156).
8. Khaksar Manshad M, Meybodi MR, Salajegheh A. A new irregular cellular learning automata-based evolutionary computation for time series link prediction in social networks. Applied Intelligence. 2021 Jan;51(1):71-84.
9. McDermott MJ, Dwaraknath SS, Persson KA. A graph-based network for predicting chemical reaction pathways in solid-state materials synthesis. Nature communications. 2021 May 25;12(1):1-2.
10. Md V, Misra S, Ma G, Mohanty R, Georganas E, Heinecke A, Kalamkar D, Ahmed NK, Avancha S. Distgnn: Scalable distributed training for large-scale graph neural networks. InProceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis 2021 Nov 14 (pp. 1-14).
11. Nasiri E, Berahmand K, Rostami M, Dabiri M. A novel link prediction algorithm for protein-protein interaction networks by attributed graph embedding. Computers in Biology and Medicine. 2021 Oct 1;137:104772.
12. Nirmala P, Nadarajan R. Cumulative centrality index: Centrality measures based ranking technique for molecular chemical structural graphs. Journal of Molecular Structure. 2022 Jan 5;1247:131354.
13. Roger R.F., Leonardo W.D., Donald J.T., “Title of Our Research Paper”, Name of the Publisher/Journal, April 2015, 7 (3), 129–151.
14. Varma S, Shivam S, Thumu A, Bhushanam A, Sarkar D. Jaccard Based Similarity Index in Graphs: A Multi-Hop Approach. In2022 IEEE Delhi Section Conference (DELCON) 2022 Feb 11 (pp. 1-4). IEEE.
15. Wang M, Qiu L, Wang X. A survey on knowledge graph embeddings for link prediction. Symmetry. 2021 Mar 16;13(3):485.
16. Xue H, Yang L, Jiang W, Wei Y, Hu Y, Lin Y. Modeling dynamic heterogeneous network for link prediction using hierarchical attention with temporal rnn. InJoint European Conference on Machine Learning and Knowledge Discovery in Databases 2021 (pp. 282-298). Springer, Cham.
17. You K, Park HJ. Re-visiting Riemannian geometry of symmetric positive definite matrices for the analysis of functional connectivity. NeuroImage. 2021 Jan 15;225:117464.
18. Yun S, Kim S, Lee J, Kang J, Kim HJ. Neo-gnns: Neighborhood overlap-aware graph neural networks for link prediction. Advances in Neural Information Processing Systems. 2021 Dec 6;34:13683-94.
19. Zhang Q, Yu K, Guo Z, Garg S, Rodrigues J, Hassan MM, Guizani M. Graph neural networks-driven traffic forecasting for connected internet of vehicles. IEEE Transactions on Network Science and Engineering. 2021 Nov 18.

20. Zhang Q, Yu K, Guo Z, Garg S, Rodrigues J, Hassan MM, Guizani M. Graph neural networks-driven traffic forecasting for connected internet of vehicles. IEEE Transactions on Network Science and Engineering. 2021 Nov 18.
21. Zheng X, Zhang L, Li K, Zeng X. Efficient publication of distributed and overlapping graph data under differential privacy. Tsinghua Science and Technology. 2021 Sep 29;27(2):235-43.