Inferring road network structure using link prediction

Ovidiu Victor Tătar

Abstract—**This paper proposes a method for determining missing roads in a road-network using link prediction. We load road-data from OpenStreetMap into Dgraph, a distributed graph database, and use established similarity-functions as well as spacial information to determine a link confidence for any given link. For this task we train random forest classifiers and measure the accuracy of our predictions on two similar datasets. We find that the trained models produce at worst predictions with an average recall-rate over 16% while maintaining an average precision of at least 52% and an overall average accuracy of over 99%.**

I. Introduction

Infrastructure and by extension road networks have a significant impact on our modern-day lives, as roads are an important part of transporting people and goods. Gaining new insights into how road networks work and how they might change over time can therefore be considered valuable information. Traditionally this problem of analyzing networks over time and predicting changes has been approached using *link prediction*.

Originally the problem of link prediction has been formalized for social networks [see 1], but has since been applied to other fields. Link prediction aims to find edges that are not included in a graph-dataset, but are likely to be true and should be included in the dataset. [see 2, p. 61] The classical approach to link prediction popularized by Liben-Nowell and Kleinberg assigns a *similarity score* to every pair of nodes in a graph using a *similarity function*. These similarity scores can be used to determine the confidence of a given link existing in a graph-dataset.

Road networks can be expressed as graphs, therefore link prediction should be applicable to aid in the deduction of new information. Possible applications include:

- generating realistic mock-up-data for other applications
- reconstructing road networks with missing roads (for example a medieval road network, that was modernized and destroyed over the centuries)
- predicting the evolution of a road network given its previous history; This could be helpful in city- and infrastructure-planning.
- reimagining a road network in the style of another road network (*style transfer* for road networks); This could be used for example to model how a rural area would evolve into a urban one or to aid in planning efficient road networks (by transferring knowledge about well designed road networks).

Each of these applications brings with them their own challenges, we mention some of these as limitations in section V. Going forward we will not further explore these different applications in this paper, instead they are a outlook for what we believe might be possible with future research.

This paper aims to apply link prediction to road networks and therefore to infer links (potential roads) not present in a given road network. Given a road network with incomplete data, the task is to predict how likely a link/road is to connect two endpoints. We propose utilizing established similarity-functions as well as spacial information to train a random forest classifier to assign a confidence for each given link.

II. Related Work

Wang, Pan, Li, *et al.* show problems of classical similarity-functions (common neighbors, jaccard, etc.) in the specific domain of road networks. The authors propose a metric based on subgraphs which shows good performance on the proposed datasets and outperforms many other metrics in prediction accuracy. [see 3, p. 20] However their proposed representation of the road network understands nodes as roads, spanning multiple physical road-sections. Two nodes share an edge if the roads intersect at any point. [see 3, p. 7] Therefore this representation does not directly allow inclusion of spacial information such as road-lengths for the prediction task, and a predicted link does not necessarily provide information on how the roads should intersect (where the intersection-point is located). Additionally our representation of the network enables us to extract our dataset using an open-source python-library (OSMnx) directly from OpenStreetMap for any location.

Julian and Lu show the effectiveness of neural networks and random forest classifiers for typical link prediction datasets. [4] We further explore the uses of random forest classifiers for link prediction in the context of road networks.

III. Approach

We use the open-source python-library OSMnx [5], which fetches road-networks for a given location directly from OpenStreetMap. OSMnx itself uses NetworkX [6], another python package for graph-network creation and analysis, which will be used to compute different similarity-scores.

The fetched road-network is transformed into a `JSON`-representation suitable to be loaded into a Dgraph-database [7]. We use Dgraph mainly as a way to permanently store different datasets, which also enables us to query the data using for example queries involving geographical locations. As a graph database, Dgraph is suitable as storage for combined data from different sources while retaining semantic information, as nodes and edges can have attributes stored directly with them. However Dgraph is not directly involved in the link prediction task and this step can be skipped for the purpose of the method this paper proposes.

The datasets are split into a test- and "training"-network by randomly assigning edges to either one of the networks or the other: The similarity functions are applied on all non-existent connections (the edges of the graph complement); then the test-network is used to evaluate the performance of the assigned scores, as it contains all possible correct predictions.

Using Scikit-learn [8] we train random forest classifiers. Random forest classifiers combine the predictions from multiple decision tree classifiers, which have been trained with random subsamples of the provided data.[(1)] Using the features observable from the training network and real connections in the original network (before the test-training split) for labeling, the classifier is trained on all connections (existing and non-existing).

In subsection III-A a more formal description of how a road network can be interpreted as a graph is provided, in subsection III-B the datasets used are described in more detail and in subsection III-C the different similarity functions used in the experiments are elaborated upon.

A. Modeling road networks as graphs

This paper models a road as a connection between two different nodes. A road network can therefore be directly modeled as an undirected graph $G = (V, E)$ with $E \subseteq \{\{x, y\} \mid x \in V \wedge y \in V \wedge x \neq y\}$ defined as the set of roads (edges) in the network and V being their endpoints (nodes). Let $\deg(x) = |\{\{x, y\} \mid y \in V\}|$ for any node $x \in V$ be the degree of said node, which can be intuitively understood as the number of roads connected to it. As an example a node a with $\deg(a) = 1$ is a dead end, a node b with $\deg(b) = 4$ is a four-way intersection and another node c with $\deg(c) = 2$ is the point where two roads meet.

However this does not reflect reality, where streets can for example lead through multiple intersections without changing identity. To model streets as they appear in real-world road networks, real streets R can be modeled as subsets of multiple edges $R \subseteq E$. Additionally one-way roads exist, which could be modeled explicitly using directed graphs. As we are not interested in modeling the exact semantics of road networks, but only their structure, going forward this paper understands roads as edges $r \in E$ of an undirected graph $G = (V, E)$.

B. Datasets

As the datasets used in this paper are accessed via OSMnx from OpenStreetMap, the underlying data is publicly available. For the experiments we chose

Table I
Properties of the datasets used in this paper

Dataset	Nodes	Edges	Clustering coefficient
tub	649	2435	≈ 0.17155
ufrank	646	2317	≈ 0.16966

two similar road networks, both located in the city of Berlin.

tub This is the road network surrounding the “Technische Universität Berlin”. More specifically this is a 3km by 3km square portion of the road network accessible by car centered at the geographical location $52.51101585, 13.326954140959668$.

ufrank This is the road network surrounding the subways station “U Frankfurter Tor”. More specifically this is a 3km by 3km square portion of the road network accessible by car centered at the geographical location $52.515818, 13.4539745$.

Both road networks are simplified in multiple steps:

1) In a process OSMnx calls *Graph simplification*, nodes are removed that are only used for road geometry. Instead the shape of a road is completely transferred to the edges.[(2)]
2) In a process OSMnx calls *Consolidation*, nodes that are part of the same intersection are merged together.[(2)]For this we chose that nodes which are up to 50 meters apart and can be considered to be part of the same intersection will be merged. We chose this number, as the datasets contain rather big circular intersections, such as the one at “Ernst-Reuter-Platz”.
3) Multiple edges between two nodes are merged together, loosing the exact road geometry.
4) Finally the network is transformed into an undirected graph by declaring all directed edges as undirected edges.

This simplification process not only aims to reduce the number of edges and nodes for the process of link prediction, but also abstracts away from the exact geometry of the road network. As can be seen in Table I the final datasets are similar in the number of edges and nodes and degree of clustering. In Figure 1 both datasets are visualized using OSMnx.

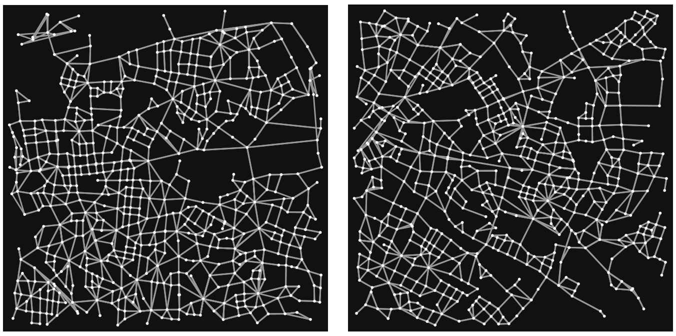

Figure 1. Visualization of datasets created with OSMnx. left: tub, right: ufrank. Map data © OpenStreetMap contributors, see notes for more information

C. Similarity scoring

We use different similarity functions to score edges, some (like common neighbors, jaccard coefficient or adamic adar index) have been used traditionally for link prediction [see 1, p. 1021] while others are less commonly used. In contrast to other applications for link prediction, road networks include spacial information that will also be used for assigning similarity scores.

Let $\Gamma(x)$ be the set of neighbors of a node x in a graph. Let $\{x, y\}$ be any edge in a graph.

common neighbors (CN) the number of neighbors the two nodes of the edge share; $|\Gamma(x) \cap \Gamma(y)|$

jaccard coefficient (JC) a normalized coefficient denoting how many neighbors the two nodes of the edge share; $\dfrac{|\Gamma(x) \cap \Gamma(y)|}{|\Gamma(x) \cup \Gamma(y)|}$

adamic adar index (AA) a coefficient denoting how many neighbors the two nodes of the edge share with diminishing returns; $\displaystyle\sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{\log(\Gamma(z))}$

min. neighbors (min) the number of neighbors of the edge’s node that has less neighbors[(3)]; $\min\{|\Gamma(x)|, |\Gamma(y)|\}$

max. neighbors (max) the number of neighbors of the edge’s node that has more neighbors[(3)]; $\max\{|\Gamma(x)|, |\Gamma(y)|\}$

geospacial x-distance (xdist) the distance of the nodes’ geographical locations regarding longitude only

geospacial y-distance (ydist) the distance of the nodes’ geographical locations regarding latitude only

geospacial distance (dist) the euclidean distance of the nodes' geographical locations

top-k shortest distance (short) Introduced by Lebedev, Lee, Rivera, *et al.* as a similarity function for link prediction, this is the sum of the lengths of the first k shortest paths between the edge's nodes. [9] The authors found that small values for k worked best in their datasets [see 9, p. 4], so we chose a value of $k = 3$. We also chose to consider the lengths of the shortest paths separately, so **short1** refers to the length of the shortest path, **short2** refers to the length of the shortest path excluding the first and **short3** refers to the length of the shortest path excluding the first two.

IV. Experiments

The road networks described in subsection III-B are split into a test- and training-network by randomly assigning edges to either network. We carry out our experiments with two different splits:

1) a 80-20 training-testing split (**tub20** and **ufrank20** respectively)
2) a 70-30 training-testing split (**tub30** and **ufrank30** respectively)

Due to limited computing resources, we have run our experiment only five times (for every dataset and splitting method) and the resulting values referenced represent the mean over these five runs.

We apply the similarity functions described in subsection III-C to all non-existent connections of the training-network and evaluate the results using the test-network.

For every function we computed the *area under the ROC-curve* (AUROC) and the results are shown in Table II. AUROC allows us to compare the performance of a function to random guessing [see 10, p. 40], as guessing randomly for each datapoint would result in a AUROC-score of 0.5.

For the computation the AUROC-score assumes, that a higher similarity score corresponds to a higher probability, that the link should exist. For some similarity functions this is not true, and instead the AUROC of their negated output was computed, such that a lower similarity score leads to a higher link probability. This has been highlighted in the referenced table with a "−" before the abbreviation of the function.

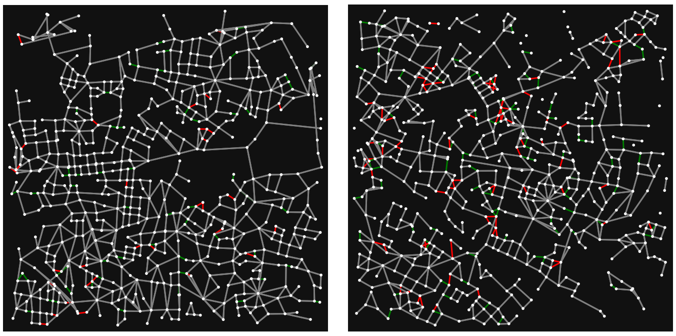

Figure 2. Visualization of predictions created with OSMnx. left: ufrank20 on tub20 small, right: tub30 on ufrank30 large. Grey edges are known (training-network), green edges represent correct guesses, red edges false ones; false negatives are not shown. Map data © OpenStreetMap contributors, see notes for more information

The random forest classifiers have been trained on one complete dataset and are being evaluated on the dataset they have not trained on. A classifier trained on "tub20" and evaluated on "ufrank20" will be referenced as "tub20 on ufrank20". We have trained classifiers with 50 decision trees and a maximum tree depth of 10 (these have the suffix "small"), and classifiers with 100 decision trees and a maximum depth of 50 (suffix "large"). The results for some common performance metrics for classifiers are shown in Table III. Finally, we have also visualized the predictions of two of the trained classifiers in Figure 2 (these are not cherry-picked).

A. Discussion

Our results show low AUROC for the similarity functions "CN", "JC" and "AA", confirming the findings of Wang, Pan, Li, *et al.* [see 3, p. 20]

In general the standard deviation is relatively low (speaking for the representativeness of the results) and the similarity functions based on graph distance and on geospacial distance show the best performance. However it should be noted, that computing the geographical distance was less time intensive. Although not shown in the results above, calculating the function "short" took on average over 400 × more time than any other function on our computing environment.

Hence concerning datasets we used or datasets sim-

Table II
AUROC FOR DIFFERENT SIMILARITY FUNCTIONS

		CN	JC	AA	−min	−max	−xdist	−ydist	−dist	−short	−short1	−short2	−short3
tub20	mean	0.635504	0.635623	0.635659	0.627206	0.581656	0.952672	0.952200	0.995198	0.812849	0.853254	0.812823	0.795867
	std.	0.013376	0.013396	0.013392	0.008794	0.018746	0.001881	0.001453	0.000636	0.033307	0.027014	0.031193	0.032282
tub30	mean	0.615754	0.615837	0.615859	0.591949	0.551532	0.953862	0.951671	0.995839	0.749292	0.799927	0.751449	0.724398
	std.	0.010361	0.010351	0.010343	0.005995	0.007744	0.001544	0.001061	0.000306	0.013059	0.012457	0.012353	0.012540
ufrank20	mean	0.632681	0.632904	0.632783	0.632304	0.586775	0.953170	0.953767	0.996590	0.782483	0.829498	0.786609	0.758246
	std.	0.013331	0.013384	0.013300	0.014338	0.015329	0.001579	0.001369	0.000451	0.024275	0.026360	0.025041	0.020442
ufrank30	mean	0.605373	0.605493	0.605448	0.609986	0.584332	0.952126	0.953461	0.996258	0.676313	0.752542	0.683670	0.646840
	std.	0.004467	0.004489	0.004487	0.008024	0.014252	0.000571	0.000949	0.000203	0.038982	0.034168	0.039458	0.035356

Table III
PERFORMANCE OF THE RANDOM FOREST CLASSIFIERS

		accuracy mean	accuracy std.	recall mean	recall std.	precision mean	precision std.
tub20 on **ufrank20**	large	0.998935	0.000025	0.189913	0.018090	0.530258	0.033606
	small	0.998945	0.000023	0.165929	0.029215	0.549297	0.035229
tub30 on **ufrank30**	large	0.998412	0.000032	0.244223	0.012845	0.524902	0.018915
	small	0.998451	0.000034	0.235323	0.017230	0.556468	0.023496
ufrank20 on **tub20**	large	0.998915	0.000023	0.237205	0.025487	0.551084	0.021606
	small	0.998925	0.000031	0.208588	0.023491	0.572757	0.034146
ufrank30 on **tub30**	large	0.998410	0.000027	0.273833	0.019809	0.568415	0.016990
	small	0.998409	0.000011	0.244656	0.018904	0.576806	0.007279

ilar to these, geospacial distance can be considered an important similarity function to consider.

While there are trends between different datasets and splits, they lie within the standard-deviation and since only the results of five experiments are being considered, we will abstain from making any conclusions from these trends.

V. Conclusion

In this paper we presented a method for predicting links using different similarity metrics and random tree classifiers. Although we have run only few experiments, on our datasets our method shows a useful recall-rate of over 16% on average while maintaining an acceptable precision of over 52%.

Our method however does not

- consider obstacles in the terrain, such as buildings or lakes,
- consider multiple roads/connections between nodes,
- produce the shape of roads or location of nodes

and has not been tested with data from possible real-world applications or rural areas.

Future work should not only focus on these limitations, but should also explore other methods of prediction on the specific domain of road networks to improve performance and test the feasibility for the real-world applications mentioned in the introduction of this paper.

Nevertheless, our paper introduced geospacial distance as a important feature to consider for link prediction in road networks and proposed a method for dataset extraction that can be applied to any real-world location using publicly available data. Finally given the performance of our classifiers, we show link prediction on road networks is feasible and thus we believe link prediction on road networks can have a interesting future.

Notes

(1) see https://scikit-learn.org/0.23/modules/ensemble.html#random-forests and https://scikit-learn.org/0.23/modules/generated/sklearn.ensemble.RandomForestClassifier.html; both last accessed on 26th September 2020

(2) see https://github.com/gboeing/osmnx-examples/blob/64e104f8e3e719c23c640172c2f18ba7b46a020d/notebooks/04-simplify-graph-consolidate-nodes.ipynb; last accessed on 26th September 2020

(3) As subsection III-A suggests, the degree (number of neighbors) of a node contains information on the type of the node (intersection, etc.)

References

[1] D. Liben-Nowell and J. Kleinberg, "The link-prediction problem for social networks," *Journal of the American Society for Information Science and Technology*, vol. 58, no. 7, pp. 1019–1031, 2007. DOI: 10.1002/asi.20591.

[2] A. Hogan, E. Blomqvist, M. Cochez, C. d'Amato, G. de Melo, C. Gutierrez, J. E. L. Gayo, S. Kirrane, S. Neumaier, A. Polleres, R. Navigli, A.-C. N. Ngomo, S. M. Rashid, A. Rula, L. Schmelzeisen, J. Sequeda, S. Staab, and A. Zimmermann, *Knowledge graphs*, 2020. arXiv: 2003.02320 [cs.AI].

[3] B. Wang, X. Pan, Y. Li, J. Sheng, J. Long, B. Lu, and F. R. Khawaja, "Road network link prediction model based on subgraph pattern," *International Journal of Modern Physics C*, 2020. DOI: 10.1142/S0129183120500837.

[4] K. Julian and W. Lu, "Application of machine learning to link prediction," 2016. [Online]. Available: http://cs229.stanford.edu/proj2016/report/JulianLu-Application-of-Machine-Learning-to-Link-Prediction-report.pdf.

[5] G. Boeing, "OSMnx: New methods for acquiring, constructing, analyzing, and visualizing complex street networks," *Computers, Environment and Urban Systems*, vol. 65, pp. 126–139, 2017, ISSN: 0198-9715. DOI: 10.1016/j.compenvurbsys.2017.05.004.

[6] A. A. Hagberg, D. A. Schult, and P. J. Swart, "Exploring network structure, dynamics, and function using NetworkX," in *Proceedings of the 7th Python in Science Conference*, G. Varoquaux, T. Vaught, and J. Millman, Eds., Pasadena, CA USA, Aug. 2008, pp. 11–15.

[7] M. Jain, "Dgraph: Synchronously replicated, transactional and distributed graph database," version 0.8, p. 11, Feb. 2020. [Online]. Available: https://dgraph.io/paper.

[8] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[9] A. Lebedev, J. Lee, V. Rivera, and M. Mazzara, *Link prediction using top-k shortest distances*, 2017. arXiv: 1705.02936 [cs.SI].

[10] D. M. W. Powers, "Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation," *Journal of Machine Learning Technologies*, vol. 2, pp. 37–63, 1 2011.