Geospatial Prediction Using Road Topology: A Graph-based Perspective

Yameng Guo[1], Seppe vanden Broucke[1,2,*]

[1]Department of Business Informatics and Operations Management, Ghent University, Tweekerkenstraat 2, 9000 Gent, Belgium
[2]Research Centre for Information Systems Engineering, KU Leuven, Naamsestaat 69, 3000 Leuven, Belgium

Abstract

A wide variety of literature has aimed to integrate the road network into spatial predictive tasks by utilising non-Euclidean distance metrics. Many of these studies have concentrated solely on the length of the path between nodes in a pairwise fashion, which disregards important topological properties and structure of the road network as a whole. Given the increasing popularity of graph-based methodologies such as Graph Neural Networks, this work reflects on the conventional approaches and proposes a graph-based perspective to incorporate road network information in the context of geospatial prediction. Specifically, we propose a unified graph structure that incorporates both target nodes and road nodes and retains both geospatial (location) and topological (structure) information. We then conduct a comparative experiment including statistical models, machine learning techniques, and graph-specific models. The results of our experiment demonstrate that representing road-topology in a graph-based manner extends the range of available techniques in contexts where roads play an important role, i.e. real estate valuation modeling. Our approach is also highly interpretable and can be easily visualized to provide insightful results.

Keywords

geospatial modeling, graph analytics, road network topology, real-estate valuation

1. Introduction

For many decades, scholars have recognized the importance of spatial relationships amongst observations in contexts where observations are associated with a particular location [1]. When tasked with the construction of a predictive model for such observations, it is well known that spatial relationships should be derived in a different manner compared to conventional (non-spatial) settings. That is, the assumption of independent and identically distributed random variables typically does not apply, as spatial information comes with dependency structures that must be taken into account.

Throughout the past years, we have thus observed a variety of specialized techniques being applied to predictive tasks in the geospatial domain, which typically require the construction of a pairwise distance matrix between observation or the definition of a kernel function to define a neighborhood around observations. Many existing applications do so by expressing the notion of proximity between observations in Euclidean terms ("as-the-crow-flies"), based on an idealized Cartesian map.

Some works have challenged this assumption, and have argued for using non-Euclidean distance [2]. Especially in urban settings, it was observed that usage of distance metrics based on travel times and road distances could lead to a significant improvement in terms of predictive power [3, 4], for instance in the context of real estate valuation (e.g. house price prediction) or rental cost estimation, i.e. targets where the road topology is understood to play an important role.

However, the usage of road-based distance measures or road networks in general comes with particular challenges. That is, many statistical techniques make assumptions in terms of covariance or variogram properties which cannot be guaranteed with non-Euclidean distance functions. As such, attempts to use such distance metrics typically apply Minkowski approximations to make the resulting metric behave more properly. Another approach, commonly applied for kernel-based techniques, is to add "barriers": obstacles that influence the shortest path between two points. However, correctly modeling road networks would require adding in barriers around every road, which negatively impacts commonly used kernel functions such as the radial basis function.

In this work, we explore an alternative mechanism to incorporate the road topology into a geospatial predictive task, namely a graph-based representation where a road network is converted into a graph consisting of vertices and edges. We introduce here an adjusted graph representation which also retains location information for vertices based on which euclidean distances can still be easily retrieved. We argue that this representation offers a best of both worlds approach, based on which both traditional statistical and machine learning meth-

STRL'23: Second International Workshop on Spatio-Temporal Reasoning and Learning, 21 August 2023, Macao, S.A.R
yameng.guo@ugent.be (Y. Guo);
seppe.vandenbroucke@ugent.be (S. vanden Broucke)
0000-0003-2719-1356 (Y. Guo); 0000-0002-8781-3906 (S. vanden Broucke)

ods, as well as graph-centric approaches can be easily applied. As such, our goal is to convey the following contributions:

- We present a graph-based representation for geospatial data which retains both location and road topology information;
- We conduct a comparative experiment including well-known statistical and machine learning techniques together with novel graph-based models adapted to our representation;
- We show how our representation leads to interpretability benefits;
- An implementation of our work is made publicly available[1].

The remainder of this work is structured as follows: Section 2 provides an overview of related work. Next, Section 3 presents our methodology towards constructing a graph-based representation for geospatial data and describes graph-centric models that can be applied on it. Section 4 then describes the experimental setup to compare a number of geospatial models, including the graph-centric models described previously. Section 5 discusses the results, and illustrates the interpretability benefits of our approach. Finally, the paper is concluded in Section 6.

2. Related Work

Traditional statistical approaches towards modeling regression tasks in a geospatial setting are grounded on exploring the correlation between geographical instances using Euclidean distance as a basis [5]. Various interpolation techniques, such as Kriging [6], geographically weighted regression (GWR) [7], and inverse distance weighting (IDW) [8] have been widely utilized in this regard.

As was elaborated in the introductory section, it is possible to utilize another distance metric (e.g. to calculate pairwise distances in the case of Kriging) or kernel function (in the case of geographically weighted regression), which in theory allows the user to break away from the Euclidean assumption. For example, [9] investigated this approach by applying a GWR model calibrated with a non-Euclidean metric that included road network distance and travel time metrics. In [3], similar metrics where utilized and combined with a Kriging model. To satisfy the underlying assumptions of this method, the authors present a Minkowski approximation to make the resulting metric well-behaved in terms of symmetry and the triangular inequality property. In [10, 4], similar adjustments were proposed, where the authors rely on embedding approaches and dimensionality reduction techniques such as Isomap to tackle the issue of the non-Euclidean distance metric leading to invalid spatial covariance. Whilst rooted in solid statistical theory, one drawback of this group of techniques is that ultimately, such approximations do remove some more intricate details resulting from the road network connecting instances. Additionally, relying on pairwise distances between observations does cause that a more global insight in terms of the road topology (or road network) is lost.

Other scholars have also focused towards exploring a graph-oriented perspective for geospatial prediction. Notably in [11], the authors argue towards an integration of network information with location information by directly leveraging the road network as a graph representation, which is most closely in line with what is presented herein. Contrary to that work, our focus lies more on fine-tuning the manner how the graph is utilized to establish predictions rather than efficient query mechanisms. Recent work such as [12] has also confirmed the viability of this approach.

Given the growing popularity of deep learning techniques that work directly on graph-based data in recent years, an increasing number of researchers are also exploring potential solutions for house price prediction by applying related techniques. For example, [13] make use of geometric learning and apply an attention mechanism in the context of real estate appraisal. [14] uses a graph neural network on a graph representation constructed using point of interest information. Nonetheless, these approaches suffer once more from a disconnect between the geospatial and graph perspective: distance-based methods abstract away the road topology and have to rely on approximations, whereas graph-based methods have been mainly centered on incorporating the instances as vertices and connecting them—likewise—based on Euclidean or road distance and have seldomly considered the notion that a direct graph structure follows from a given road topology. Notably, this disconnect is also apparent from an interpretability angle, e.g. when visualizing results of a given predictive model, where most displays exhibit e.g. predictions over distinct segments on the road network without being able to provide more granular predictions over, say, a long singular street segment.

In this work, we take a new look at the graph-oriented perspective and propose an adjusted representation which shows that both road topology as well as location information can be easily incorporated in a singular view. Based on this representation, we show that novel graph-centric approaches can be easily applied by means of a comparative experiment. We also show that our representation leads to interpretability benefits by means of naturally-understandable visualization and retaining the

[1] See: https://github.com/ArmonGo/knnroadgraph

ability to make predictions on a point level rather than street or street-segment level only.

3. Geospatial Graph Methodology

In this section, we describe the two key contributions of this work. First, we detail our graph-based representation and how it can be constructed given a set of spatial observations and a road network. Second, we describe an adjusted k-Nearest Neighbor approach as a graph-centric technique that leverages this representation.

3.1. Representation and Construction

A graph is a structure over a set of objects detailing how pairs of objects are in some sense "related". The objects are typically denoted as vertices (or nodes) whereas the pairs are denoted as edges (or links, arcs). Formally, a graph is a pair $G = (V, E)$ with V the vertices and $E \subseteq \{(v_i, v_j, w) \in V^2 \times \mathbb{R}_0^+ \wedge v_i \neq v_j\}$, where $v_i, v_j \in V$ and w is the weight of the edge. Each element $v \in V$ is defined as a tuple (x, y, t, p) with x and y spatial coordinates, $t \in \{road, target\}$ indicating the vertex type (road segment node or node carrying a target value) and $p \in \mathbb{R}$ the target value (i.e. estate price). Only target nodes carry target values, so $\forall (x, y, t, p) \in V \wedge t = road : p = \emptyset$, and target nodes need to be connected to a road node, so $\forall (v_i, v_j, w) \in E : \neg(type(v_i) = target \wedge type(v_j) = target)$, with $type$ a function returning the type of a vertex.

The end result is a graph structure with two node types, geospatial coordinates for all nodes, target values for one node type, and edges inter-connecting road nodes as well as target nodes to the road structure. This structure is sufficient to present a straightforward view of a graph and geospatial representation. Note that traditional techniques can simply utilize the geospatial coordinates and targets the vertices. Also, we define d_{ij} to represent the length of geodesic path between $v_i, v_j \in V$, the geodesic path being the shortest path between two vertices utilizing the edge weights, e.g. using Dijkstra's algorithm [15].

We now detail the construction of such a graph given a series of instances with coordinates and associated target values, e.g. house prices. We also assume access to the road topology or geometry[2]. Common geospatial notation describes roads as a set of "LineString" objects (a sequence of points) or "MultiLineString" objects (a collection of the former though with no guarantees on disjointedness amongst the members), i.e. through the "Well-Known Text" (WKT) markup language for representing vector geometry objects, a format originally defined by the Open Geospatial Consortium (OGC) and now by the ISO/IEC 13249-3:2016 standard [16]. For sake of brevity, we do not formalise these objects in detail, but will assume simply a set of roads segments as a set of $R = \{(x_1, y_1, x_2, y_2)\}$ lines, where $(x_1, y_1), (x_2, y_2)$ are the corresponding coordinates for the two points defining the line.

[2]E.g. through OpenStreetMap as is provided in our reference implementation.

Algorithm 1 Geospatial graph construction from road lines and target points

Input: $R = \{(x_1, y_1, x_2, y_2)\}$ set of road lines
Input: D_x, D_y, y_t vectors of target coordinates and values
Output: $G = (V, E)$ a graph with road and target nodes

Add road nodes and edges
for $(x_1, y_1, x_2, y_2) \in R$ **do**
  $v_i := (x_1, y_1, road, \emptyset)$
  $v_j := (x_2, y_2, road, \emptyset)$
  $V := V \cup \{v_i, v_j\}$
  $E := E \cup (v_i, v_j, eucl(v_i, v_j))$
end for

Merge road nodes
for $v_i \in V$ **do**
  $C := \{v_j \in V | v_i \neq v_j \wedge eucl(v_i, v_j) < \epsilon\}$
  Remove edges from E between v_i and a $v_j \in C$
  Reconnect edges in E: replace endpoint $v_j \in C$ with v_i
  $V := V \setminus C$
end for

Add target nodes and edges
for $(x, y, p) \in ((D_x, D_y) \times y_t)$ **do**
  $v_t := (x, y, target, p)$
  Find closest edge $e_c = (v_i, v_j, w) \in E$ to v_t with both endpoints road nodes
  Find closest point (x_c, y_c) along line of edge e_c to v_t
  $v_c := (x_c, y_c, road, \emptyset)$
  $V := V \cup \{v_t, v_c\}$
  $E := E \cup \{(v_i, v_c, eucl(v_i, v_c)), (v_c, v_j, eucl(v_c, v_j))\}$
  $E := E \cup \{(v_t, v_c, eucl(v_t, v_c))\}$
  $E := E \setminus \{e_c\}$
end for

The construction of the graph representation is outlined in Figure 1 and further detailed in Algorithm 1. The first step consists of defining the main road topology by adding in road nodes and edges (*eucl* is returns the Euclidean distance between two vertices using their spatial location). This is, in practical terms, not directly ready for use, i.e. numerical stability inconsistencies as well as measurement errors typically lead to a graph with many disconnected edges. Hence, the set of road vertices are merged based on their spatial location as described in [17] (ϵ here is a small threshold value). Next, target nodes are added in and connected to the road graph. For the sake of computational efficiency, one could opt here to add in edges so that each target node is connected to their closest road node directly. Given the distance-based edge weights which are used to derive geodesics, it is

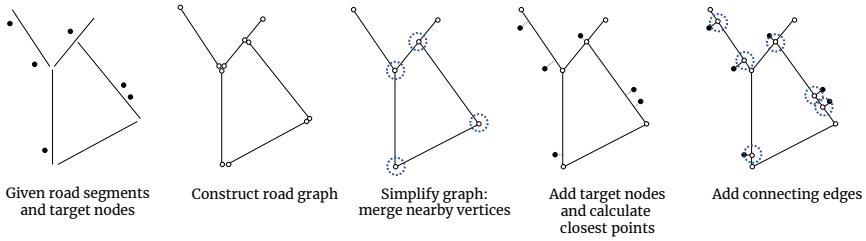

Figure 1: Overview of the graph construction procedure given a set of target nodes and road segments

more appropriate to connect target nodes based on spatial closeness to the road network, as we cannot make strong assumptions in terms of granularity of road segmentation. Hence, we connect each target node to their nearest found points on the road network (spatially) by creating a road node at that nearest location, add in an edge to the target node, add in two edges from that road node to the original endpoints of the nearest edge (also road nodes), and remove the original edge.

3.2. Models

To emphasize the efficacy our graph-centric representation, we present a series of modifications for the well known k-Nearest Neighbor (K-NN) algorithm [18] to construct four different variations that will be contrasted to alternative techniques in the next section.

In general, K-NN works as follows: given a set of training instances $(X_1, y_1), (X_2, y_2), \dots, (X_n, y_n)$ each $\in \mathbb{R}^d \times \mathbb{R}$, where y is the target value of X, the feature vector describing instances. Given a distance function $d(i, j) : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ and a given point $P \in \mathbb{R}^d$, let $(X_{(1)}, y_{(1)}), (X_{(2)}, y_{(2)}), \dots, (X_{(n)}, y_{(n)})$ be the reordering of the training instances so that $d(X_{(1)}, P) \leq \dots \leq d(X_{(n)}, P)$. Given a value $k \geq 1$, predictions for new points can be made as follows:

$$\hat{y}(P) : \mathbb{R}^d \to \mathbb{R} = \sum_{i=1}^{k} w_{(i)} \times y_{(i)} \tag{1}$$

where $w = \langle w_{(1)}, \dots, w_{(k)} \rangle$ a vector weighting the neighbors. By default, w is uniform so that $w_i = \frac{1}{k}$ for each i. Often, the inverse distance is also used, i.e. $w_i = \frac{\sum_{i=1}^{k} d(X_{(i)}, P)}{d(X_{(i)}, P)}$ (a small constant can be added to prevent division by zero; alternatively, in the presence of zero distances, $w_i = 1$ if $d(X_{(i)}, P) = 0$ or 0 otherwise).

Similar to [11], we will use as a distance function the length of the geodesic path on the constructed graph, so that

$$d(x_1, y_1, x_2, y_2) : \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R} = d_{v_l(x_1,y_1),v_l(x_2,y_2)} \tag{2}$$

Table 1
Different variations of the proposed K-NN approach

Name	Decay	Minmax	Weighting function
KNN-Power	✗	✗	Power inverse
KNN-PowerDecay	✓	✗	Power inverse
KNN-Softmax	✗	✓	Softmax
KNN-SoftmaxDecay	✓	✓	Softmax

with d_{ij} returning the length of the geodesic path between two nodes $i, j \in V$ as introduced earlier, the geodesic path being the shortest path between two vertices utilizing the edge weights and $v_l(x, y) = \bigcup\{v \in V | type(v) = target \wedge x(v) = x \wedge y(v) = y\}$ returning the node positioned at a given location. For instances present in the given training set, it is guaranteed by construction that they can be found in the graph. However, given a new position P, it is likely no matching node can be found. Two approaches are possible to resolve this. First, if the set of nodes to be predicted on during inference is known, they can be simple added in as target nodes with their target set blank. They are then to be ignored during the retrieval of neighbors (as neighbors should only consist of nodes considered to be in the train set). This procedure allows that the full graph and distances can be pre-calculated and is hence more computationally efficient. In case inference over unknown nodes is necessary, one has to modify the graph at inference time, i.e. add P as an target node to the graph, find the closest connection point, add it as a road node to the graph, add an edge to connect the node to the graph, add two edges from the connection point to the original endpoints of the closest edge, remove the original edge, and then infer the closest neighbors only using target nodes different from P.

Next, we introduce the following methods to establish the weight vector w_i. First, weights are set equal to the retrieved distances $w_i = d(X_{(i)}, P)$. Naturally, they do not yet sum to one. First, a decay effect can be optionally applied as such:

$$w_i = 1 - \exp(-\alpha * w_i) \tag{3}$$

with α the decay amount. Afterwards, minmax scaling can be applied so that

$$w_i = \frac{w_i - min(w)}{max(w) - min(w)} \tag{4}$$

Next, to establish the final weights, one of both weighting functions has to be applied. The first choose is to apply a power inverse weighting:

$$w_i = (\frac{max(w)}{w_i})^{\beta} \tag{5}$$

with β a hyperparameter to adjust the distribution of neighboring distances. Alternative, a softmax weighting can be applied:

$$w_i = \exp(\frac{-w_i}{\sigma}) \tag{6}$$

with σ a hyperparameter to adjust the distribution of neighboring distances. Finally, the weights are normalized so that:

$$w_i = \frac{w_i}{\sum_{i=1}^{k} w_i} \tag{7}$$

Not all scaling steps are necessary depending on the weighting function chosen: four variations of the steps above are viable and are summarized in Table 1.

4. Experimental Setup

In this section, we describe an empirical experiment to compare our presented K-NN based models with other well-known geospatial predictive modeling techniques, using a collection of real-life datasets.

4.1. Datasets

Four different datasets were collected, encompassing different regional location and predictive tasks across the country of Belgium. To retrieve the road network, official and open road data provided by the Flemish geospatial institute[3]. Regarding instances, properties where sourced for the cities of Brussels and Antwerp respectively. An open Airbnb dataset[4] was used to provide a rental-based regression target (price per property per person per night per num. bedrooms). We also obtained properties with a valuation based sale regression target (i.e. house price per square meters) for both cities from a Belgian real estate platform.

Table 2 describes the different datasets and properties of the resulting graph after conversion. Each dataset cleaned to remove duplicate values and was then partitioned into a training set (60%), a validation set (20%) and a test set (20%). Furthermore, transformations were applied to the coordinate systems describing property locations and road locations in order to align them to the same Cartesian coordinate system (EPSG:31370). The datasets are also visualized in Figure 2. Note that the sale-based datasets exhibit an even distribution across the map, whilst the rental-based data is more concentrated.

[3]See: https://www.vlaanderen.be/datavindplaats/
[4]See: http://insideairbnb.com/get-the-data/

Table 2
Different datasets used in the experiment and properties of the constructed graphs

Name	Region	Target	# Target nodes	# Road nodes	# Edges
A-S	Antwerp	Sale price	1190	10265	14846
A-R	Antwerp	Rental price	1964	11026	16382
B-S	Brussels	Sale price	415	17571	26095
B-R	Brussels	Rental price	4256	21327	33660

4.2. Models

We include the following models in our comparison. First, GWR [7] and Kriging [6], both traditional geospatial statistical techniques, using Euclidean distances for both and GWR fitting an intercept only using an adaptive kernel. We also include K-means [19], by first clustering based on the (normalized) geospatial coordinates using Euclidean distance and then using the average target per cluster to predict on unseen observations. K-means++ was used as the initialization scheme. A standard CART regression tree ("Reg. Tree") [20] as well as Random Forest [21] was also considered, as well as default K-NN

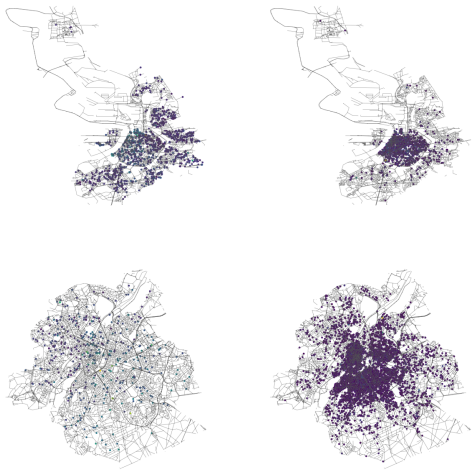

Figure 2: Visualization of the different graphs. Layout is performed using each node's spatial coordinates. Target nodes are colored according to their target value (lighter is more expensive). Rows: Antwerp, Brussels. Columns: sale, rental

using Euclidean distances with uniform and inverse distance weights ("KNN-EuclUni" and "KNN-EuclInv") [18]. With our graph representation, we include the four K-NN adaptations presented above as well as another graph-based technique, namely Label Propagation [22], a semi-supervised algorithm that assigns labels to unlabeled nodes in a graph by spreading labels from labeled nodes. Since Label Propagation does not guarantee proper convergence for regression tasks, we first discretize the target by binning it and using the average price per bin as the target when predicting on new observations.

Hyperparameters for all models were systematically tuned on the validation set using mean absolute error (MAE) before deciding upon a final model which was then evaluated on the unseen test set. Evaluation results are reported using MAE and root mean square error (RMSE). The full hyperparameter tuning grid is provided in Table 3. All experiments were executed using Python and the `scikit-learn`, `PyKrige`, `mgwr`, and `PyTorch Geometric` packages.

Table 3
Overview of the hyperparameter grid for the experimental setup. Non-listed hyperparameters were kept to their default values

Model	Hyperparameter	Range
GWR	bandwidth	$[min_{dist}/2, ..., max_{dist} \times 2]$
Kriging	num. lags	[5, 6, ..., 100]
K-means	k	[1, 2, ..., 20]
Reg. Tree	*alpha* (pruning)	[0.0, 0.1, 0.2, ..., 2.0]
Random Forest	max. feats	[0.1, 0.2, ..., 1.0]
KNN-EuclUni	k	[1, 2, ..., 20]
KNN-EuclInv	k	[1, 2, ..., 20]
KNN-Power	k	[1, 2, ..., 20]
	β (power)	[0, 0.5, ... 3]
KNN-PowerDecay	k	[1, 2, ..., 20]
	β (power)	[0, 0.5, ... 3]
	α (decay)	[0, 0.1, ... 1]
KNN-Softmax	k	[1, 2, ..., 20]
	σ (denom.)	[0, 0.1, ... 2]
KNN-SoftmaxDecay	k	[1, 2, ..., 20]
	σ (denom.)	[0, 0.1, ... 2]
	α (decay)	[0, 0.1, ... 1]
Label Propagation	num. bins	[2, 3, 4, 5]
	bin. strategy	[uniform, quantile, kmeans]
	num. layers	[1, 5, 10]
	alpha	[0, 0.1, ..., 1.0]

5. Discussion

In this section, we present the results of our experiments and highlight interpretability benefits offered by our approach.

5.1. Results

The results of the experiments on the four datasets are provided in Table 4. With regards to MAE, we notice that the graph-centric methods (bottom half in the table) outperform the traditional geospatial and machine learning techniques for three out of four datasets. With regards to RMSE, the results are more in favor of traditional techniques though here too we notice that one of the graph-based K-NN techniques obtains the advantage.

Models leveraging the graph structure hence show an interesting alternative compared to those based on location only with promising results. These findings highlight the importance of incorporating both location and topological structure and suggest the potential for utilizing graph structures in predictive modeling tasks. We must, however, highlight the various limitations still present. First of all, executing the experiment over more datasets and over repeated train-validation-test splits (e.g. by means of a more complete (nested) cross-validation) would allow to make stronger statements by means of e.g. significance tests. Also, only geospatial coordinates were used as regressors. For e.g. house valuation tasks, it is common to set up hedonic pricing models where location information is combined with other attributes describing characteristics of the property. Merging this with our graph based representation leads to interesting avenues for future work which will be discussed in the concluding section.

5.2. Interpretability

In many settings, it is important for constructed predictive models (geospatial or otherwise) to be interpretable. For instance, geospatial models are often used to make decisions that can have significant impacts on the environment, public health, and the economy. For geospatial models, an obviously appealing avenue towards getting insights in predictions offered by the model is to visualize the results on a map. Unlike previous visualization methodologies that rely on community segmentation or heatmaps, our approach quite naturally leads to a visualization where the road network is used in a direct manner.

Given a graph as detailed above and one of the K-NN based models, a visualization can be set up by interpolating across each edge and querying the model for its prediction for this new point. Note that this potentially leads to a large amount of predictions to be generated, which can be computationally expensive. As such, we also provide an approximation where predictions are generated for each road node present in the graph, which is then used to interpolate values across each edge. Visualizations for the four datasets are provided by Figure 3, overlaid on a terrain background. As can be observed, central areas of both cities display higher prices compared to the suburban areas. In Brussels, the South-Eastern part has higher prices (as is known), while in Antwerp, the highest prices spread out from the central city along the

Table 4
Experimental results. Test set MAE and RMSE evaluation results are shown for each model and dataset. The best performing model for each dataset and error metric is indicated in bold.

Evaluation	**MAE**				**RMSE**			
Models	A-S	A-R	B-S	B-R	A-S	A-R	B-S	B-R
GWR	644.87	37.56	862.33	42.95	894.83	57.50	1137.81	**295.46**
Kriging	714.26	35.68	858.32	42.14	1002.26	55.53	1133.16	295.57
K-means	688.25	35.19	902.83	42.54	927.89	**55.01**	1169.69	295.95
Reg. Tree	838.72	39.63	947.18	43.84	1150.33	75.15	1385.92	298.61
Random Forest	661.04	34.67	823.75	44.28	897.68	63.33	1162.65	296.91
KNN-EuclUni	635.16	42.13	967.61	44.83	887.88	126.52	1376.47	297.23
KNN-EuclInv	624.77	45.94	**822.04**	45.67	**851.91**	179.29	1131.50	297.84
KNN-Power	**618.67**	35.58	845.25	45.16	856.29	63.01	1133.21	297.48
KNN-PowerDecay	643.99	35.58	862.36	45.16	895.48	63.01	**1128.03**	297.48
KNN-Softmax	722.99	67.80	1063.47	57.43	989.49	427.43	2452.17	322.27
KNN-SoftmaxDecay	726.27	68.57	1070.86	57.93	996.93	429.33	2481.81	323.45
Label Propagation	857.03	**33.09**	888.70	**39.73**	1262.81	56.39	1254.80	297.18

river, indicating a discernible trend. Also interesting is the variation between the rental and sale targets. Streets located near tourist destinations, universities, and embassies demonstrate the higher prices in the rental map, whilst on the sale map, the prices of streets exhibit a smoother tendency.

6. Conclusions

The primary aim of this work was to showcase the possibility of representing road networks and location-bound observations as graphs and to examine the potential usefulness of graph-based techniques in predictive modeling tasks. A comparative experiment was conducted to illustrate the predictive and interpretative capabilities of graph-oriented methods.

The results provide some initial insights towards utilizing graph-based representations and techniques which retain both location information (coordinates) as well as topological information (road-house network structure). The limitations should also be acknowledged: an exhaustive study should consider more datasets and modeling techniques, stemming from different regions exhibiting differing population densities (our datasets stemmed from dense urban areas). This, together with a repeated experiment, would allow to make stronger statements using statistical hypothesis testing. Moreover, our setup was limited to comparatively small datasets. When considering large datasets and graphs, effort should be spend on the inclusion of efficient graph construction and neighbor retrieval mechanisms such as outlined in [23, 24, 25, 26].

Nevertheless, we do believe that there is strong potential for future work along this direction. For instance, the graph structure has not yet been fully exploited: feature engineering based on e.g. centrality metrics could be an effective strategy to easily enhance model performance. Currently, only geospatial coordinates were used as regressors. For e.g. house valuation tasks, also attributes describing characteristics of the property are typically available. Merging this with our graph based representation also leads to avenues towards future work, such as leveraging geometric learning based approaches based on message passing over graphs.

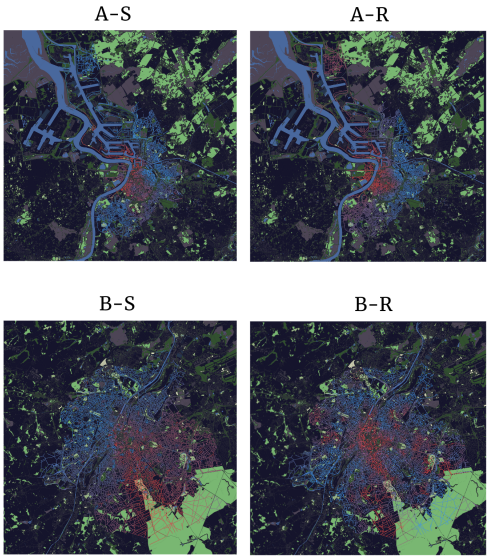

Figure 3: Visualizations of the K-NN model over the graph, overlaid on a terrain background (source: OpenStreetMap). Red represents higher predicted target values.

References

[1] W. R. Tobler, A Computer Movie Simulating Urban Growth in the Detroit Region, Economic Geography 46 (1970) 234–240. Publisher: [Clark University, Wiley].

[2] F. C. Curriero, On the Use of Non-Euclidean Distance Measures in Geostatistics, Mathematical Geology 38 (2006) 907–926.

[3] H. Crosby, T. Damoulas, A. Caton, P. Davis, J. Porto de Albuquerque, S. A. Jarvis, Road distance and travel time for an improved house price Kriging predictor, Geo-spatial Information Science 21 (2018) 185–194.

[4] H. Crosby, T. Damoulas, S. A. Jarvis, Embedding road networks and travel time into distance metrics for urban modelling, International Journal of Geographical Information Science 33 (2019) 512–536.

[5] S. Basu, T. G. Thibodeau, Analysis of Spatial Autocorrelation in House Prices, The Journal of Real Estate Finance and Economics 17 (1998) 61–85.

[6] G. Wahba, Spline Models for Observational Data, Society for Industrial and Applied Mathematics, Philadelphia, 1990.

[7] C. Bitter, G. F. Mulligan, S. Dall'erba, Incorporating spatial variation in housing attribute prices: a comparison of geographically weighted regression and the spatial expansion method, Journal of Geographical Systems 9 (2007) 7–27.

[8] University of Salzburg, A Spatial Analysis of House Prices in the Kingdom of Fife, Scotland, in: GI_Forum 2014 – Geospatial Innovation for Society, Austrian Academy of Sciences Press, Salzburg, 2015, pp. 125–134.

[9] B. Lu, M. Charlton, P. Harris, A. S. Fotheringham, Geographically weighted regression with a non-Euclidean distance metric: a case study using hedonic house price data, International Journal of Geographical Information Science 28 (2014) 660–681.

[10] H. Zou, Y. Yue, Q. Li, A. G. Yeh, An improved distance metric for the interpolation of link-based traffic data using kriging: a case study of a large-scale urban road network, International Journal of Geographical Information Science 26 (2012) 667–689.

[11] D. Papadias, J. Zhang, N. Mamoulis, Y. Tao, Query processing in spatial network databases, in: Proceedings 2003 VLDB Conference, Elsevier, 2003, pp. 802–813.

[12] T. Abeywickrama, M. A. Cheema, D. Taniar, K-nearest neighbors on road networks: a journey in experimentation and in-memory implementation, arXiv preprint arXiv:1601.01549 (2016).

[13] C.-C. Li, W.-Y. Wang, W.-W. Du, W.-C. Peng, Look Around! A Neighbor Relation Graph Learning Framework for Real Estate Appraisal, 2022. ArXiv:2212.12190 [cs].

[14] J. Du, Y. Zhang, P. Wang, J. Leopold, Y. Fu, Beyond Geo-First Law: Learning Spatial Representations via Integrated Autocorrelations and Complementarity, in: 2019 IEEE International Conference on Data Mining (ICDM), 2019, pp. 160–169. ISSN: 2374-8486.

[15] E. W. Dijkstra, A note on two problems in connexion with graphs, in: Edsger Wybe Dijkstra: His Life, Work, and Legacy, 2022, pp. 287–290.

[16] J. Melton, A. Eisenberg, Sql multimedia and application packages (sql/mm), ACM Sigmod Record 30 (2001) 97–102.

[17] D. H. Douglas, T. K. Peucker, Algorithms for the reduction of the number of points required to represent a digitized line or its caricature, Cartographica: the international journal for geographic information and geovisualization 10 (1973) 112–122.

[18] T. Cover, P. Hart, Nearest neighbor pattern classification, IEEE transactions on information theory 13 (1967) 21–27.

[19] J. A. Hartigan, M. A. Wong, A k-means clustering algorithm, JSTOR: Applied Statistics 28 (1979) 100–108.

[20] L. Breiman, J. H. Friedman, R. A. Olshen, C. J. Stone, Classification and regression trees, 1984.

[21] L. Breiman, Random forests, Machine Learning 45 (2001) 5–32.

[22] Z. Xiaojin, G. Zoubin, Learning from labeled and unlabeled data with label propagation, in: Tech. Rep., Technical Report CMU-CALD-02–107, Carnegie Mellon University, 2002.

[23] J. Sankaranarayanan, H. Alborzi, H. Samet, Efficient query processing on spatial networks, in: Proceedings of the 13th annual ACM international workshop on Geographic information systems, GIS '05, Association for Computing Machinery, New York, NY, USA, 2005, pp. 200–209.

[24] C. Shahabi, M. R. Kolahdouzan, M. Sharifzadeh, A Road Network Embedding Technique for K-Nearest Neighbor Search in Moving Object Databases, GeoInformatica 7 (2003) 255–273.

[25] H. Samet, J. Sankaranarayanan, H. Alborzi, Scalable network distance browsing in spatial databases, in: Proceedings of the 2008 ACM SIGMOD international conference on Management of data, SIGMOD '08, Association for Computing Machinery, New York, NY, USA, 2008, pp. 43–54.

[26] D. Papadias, J. Zhang, N. Mamoulis, Y. Tao, - Query Processing in Spatial Network Databases, in: J.-C. Freytag, P. Lockemann, S. Abiteboul, M. Carey, P. Selinger, A. Heuer (Eds.), Proceedings 2003 VLDB Conference, Morgan Kaufmann, San Francisco, 2003, pp. 802–813.