A Vector-Geometry Based Spatial kNN-Algorithm for Traffic Frequency Predictions

Michael May, Dirk Hecker, Christine Körner, Simon Scheider, Daniel Schulz
Fraunhofer IAIS, Schloss Birlinghoven, D-53754 Sankt Augustin, Germany
{firstname.name}@iais.fraunhofer.de, simon.scheider@web.de

Abstract

We introduce s-kNN, a nearest neighbor based spatial data mining algorithm. It belongs to the class of vector-geometry based algorithms that reason on complex spatial objects instead of point measurements. In contrast to most methods in this class, it does on the fly spatial computations that cannot be replaced by a preprocessing step without sacrificing efficiency. The key is a partial evaluation scheme for efficient computations. The algorithm is fully integrated into an object-relational spatial database. It is the basis for traffic frequency predictions (vehicles and pedestrians) for all German cities larger than 50,000 inhabitants and is the basis for pricing of posters in Germany.

1. Introduction

The *Fachverband Außenwerbung* (FAW) is the governing body of German outdoor advertising, a market with a yearly turnover of $1,200 million. FAW provides performance indicators on which the pricing of each poster is based. The performance of each poster is characterized by two measures: the number of passing vehicles, pedestrians, and public transport; and a measure which characterizes the visibility conditions of a poster, e.g. distance and orientation towards the street. Therefore, pricing of posters requires knowledge of *traffic frequencies*.

In this paper we describe a data mining algorithm that infers those traffic frequencies. For each street segment in all German cities with more than 50,000 inhabitants (approx. 1.3 Mio. segments), the model predicts the average number of cars and pedestrians per hour. This set of predictions we call a *frequency map.*

Since all poster prices in this business sector ultimately depend in an important part on the data mining algorithm described in this paper, we have a very rare if not unique case where a single data mining model is business critical for a whole branch of industry.

Our algorithm, s-kNN, is a nearest neighbor based spatial data mining algorithm. It operates on vector-geometries and can therefore reason on complex spatial data structures instead of point measurements. In contrast to most available methods in this class, it does on the fly spatial computations that cannot be replaced by a pre-processing step without loss of efficiency. The implementation is based on a partial evaluation scheme to permit efficient computation.

The rest of the paper is organized as follows. Section 2 discusses related approaches. Section 3 describes the application problem. Section 4 explains the algorithm. Section 5 gives an experimental evaluation. We conclude with a summary in section 6.

2. Spatial data mining approaches

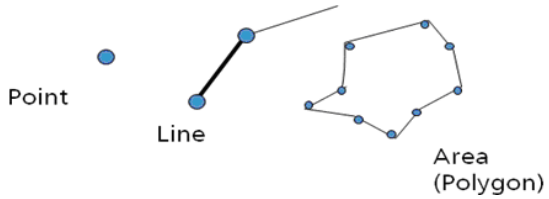

Figure 1. Types of vector geometry data: points, lines and polygons

2.1. Vector based geometries

Spatial data mining aims at the extraction of knowledge from data sets containing non-trivial spatial information [11]. We are interested here in methods that can represent extended, irregularly shaped objects, especially streets, but also houses, rivers, lakes. An object is represented as an ordered set of x-y coordinates (fig. 1). Data is normally organized in *layers* [1]. A layer describes characteristics of spatial objects including their shape and location. The geographic information systems (GIS) terminology of layers translates in data mining terminology to a *multi-relational data set with complex structured objects as values of attributes*. Most data mining algorithms assume a single matrix or relation and can deal only with numeric or string data and are not applicable to this type of da-

978-0-7695-3503-6/08 $25.00 © 2008 IEEE
DOI 10.1109/ICDMW.2008.35

ta. Multi-relational data mining [7,8] dispenses with the single table assumption, typically using a *first-order logic* representation. It does not necessarily allow for complex structured objects as values of attributes. For s-kNN, we have chosen an *object-relational* representation, since this conforms to standard vector geometry based representations in GIS.

Many implicit features can be extracted from this kind of data. Links between relations do not only consist in the usual join conditions of relational algebra, but involve complex *spatial predicates*: Features can be extracted based on geometric configurations (the 9-intersection model [3]) and proximity of objects. Other useful methods for feature extraction are buffering and spatial aggregation [1]. Feature extraction is often computationally expensive and requires considerable knowledge about geographic information systems.

2.2. Data mining approaches

A number of vector-based spatial data mining algorithms have been proposed in the literature for association rules [6,7], decision trees [6], model trees [8], density based clustering [12], and subgroup mining [5].

Sometimes a 2-step procedure is applied, where spatial data is extracted from a spatial database and converted into a first order representation as a preprocessing step [7]. The most interesting option is, however, *dynamic calculation*. Here the mining algorithm itself has the ability to perform spatial operations on the fly [5,8] as part of the search strategy. Dynamic calculation is highly attractive from the point of view of knowledge representation. The main advantage is that only those regions of the search space need to be explored that are likely to contain interesting hypotheses. *Pruning techniques* can thus avoid to perform expensive but unnecessary spatial computations [5].

In this paper we describe a dynamic approach, where the calculation is *irreducibly* dynamic. A kNN algorithm does evaluation only at run-time. Pre-calculating all spatial relations would be a very costly choice (also in terms of storage), since pre-calculation needs n x m pairs to be stored and retrieved when needed. The key to an efficient algorithm here is a *partial evaluation* of the distance function at run-time.

2.3. Autocorrelation

One important feature of spatial data is autocorrelation. Mainstream data mining is based on the assumption that data is independent and identically distributed (iid). Both assumptions may fail. If independence fails, data become dependent or autocorrelated. The autocorrelation function describes the correlation between the process at different times (temporal autocorrelation) or at different locations (spatial autocorrelation) [2].

While autocorrelation is at the heart of statistical approaches such as Kriging [2], in the class of vector-geometry based algorithms, it is often not explicitly addressed. Instead it is assumed that autocorrelation is explained by *other* attributes. An exception is the model tree algorithm Mrs-Smoti [8], which can do regression using spatial auto-correlation. We will see in section 4 how s-kNN can account for autocorrelation.

3. Application and data sources

3.1. Goal

The goal of the application can be stated thus: *Given* a number of traffic measurements and background data – a street network, socio-demographic data, and points of interest, *predict* for each street segment in every German city > 50,000 inhabitants the average number of vehicles and pedestrians per hour.

The *output* of the data mining is a set of predictions. It contains the street network and specifies for each segment the average number of vehicles and pedestrians. Using a GIS, the frequencies can be displayed as a thematic map on a street network; hence the term *frequency map* (for online examples see under https://www.faonline.de/mapsphere/faonline/). In this paper we focus on car frequencies. The prediction of pedestrian frequencies, based on the same general approach, is described in [9].

3.2. Data characteristics

The input data comprises several sources of different quality and spatial resolution.

Video measurements. For a subset of street segments frequency counts derived from video data are available, in total around 100,000 measurements for Germany. For some cities more than 2,000 measurements exists, while for others only a few dozens.

Measurements at a poster have been taken at 4 different days and 4 different times. Each measurement lasts 6 minutes. The number of cars and pedestrians has been counted manually. Reliance on automatic traffic counts is not possible because long-term traffic counts (at least in Germany) exist mostly outside the city centers, on large streets and for vehicles, while posters are located inside the cities, often on small streets and pedestrian areas. Daily average values are estimated using load-curves derived from long-term measurements. These measurements had been collected by FAW over several years and were not specif-

ically collected for the purpose of building a data mining model.

For validation purposes we have compared video measurements with long-term traffic counts made by the federal state at number of locations where such measurements coexist. The correlation is very high (0.97), demonstrating that this kind of measurements can give accurate data for the purpose at hand.

Street networks. The primary objects of interest are street segments, which are parts of a street between two intersections. Each segment possesses a geometry object and has attached information about the type of street, name of street, direction, speed class, and length. We use the Navteq network, which consists for Germany of more than 6 Mio segments. In addition, demographic and socio-economic data about the vicinity is used. It usually exists for official districts like post code areas.

Points of interest (POI). POI mark attractive places like railway stations or restaurants. Clearly, areas with a high density of restaurants will be more frequented than quiet residential areas. In order to utilize POI, data must be aggregated. Buffers were created around each street segment to calculate the number of relevant POI within the neighborhood.

4. s-kNN

We use a modified k-nearest neighbor algorithm to predict traffic frequencies. It incorporates vector spatial and non-spatial information based on the definition of an appropriate distance function. It allows to account for autocorrelation, and to bring in background knowledge (details on the last point are omitted in this paper, but can be found in [9]).

Nearest neighbor algorithms are well suited for the current task, because they do not depend on assumptions that are likely to be violated by the data set anyway. Predictions are local and therefore react robustly to outliers: a local outlier does not have global effects on the prediction.

For the following discussion it is important to distinguish between nearest neighbors in 2-dimensional geographic space and nearest neighbors in the *n*-dimensional attribute space of s-kNN. The geographic space is a *subcomponent* of the attribute space.

4.1. Feature selection

Due to the large amount of available attributes, feature selection forms an important task in our application. Consistently with the literature on kNN, early experiments showed that including a large number of attributes does not have a positive effect on the result. We used domain knowledge and exploratory data analysis for feature extraction. Relevant features for our application are the number of hotels and restaurants, number of bus stops and train stations, number of public buildings, the street category and type (main road, …). We also tried automatic subset selection methods as provided by Weka [13]. The attribute sets selected were partially overlapping with our manually selected features, however they did not perform as well during evaluation (sec. 5). A judicious choice of attributes is an important way to bring in background knowledge.

4.2. The need for vector geometries

The prediction is based on streets segments carrying frequency information (fig. 2). Segments are represented as polylines and can have complex shapes. To account for spatial autocorrelation, the spatial distance between segments needs to be calculated. In particular, two segments that meet at an intersection must have zero distance.

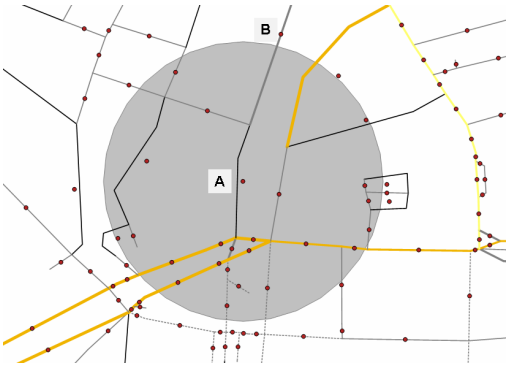

Figure 2. Centroid distances do not capture the distances between polylines well.

Simple Euclidian distance is defined between points. In case of street segments, the centroid is often taken as a point representation. This can lead to highly misleading results, as shown in fig. 2. In the street network in fig. 2, the segment with centroid A and the segment with centroid B meet each other; in fact they are the same street, and their distance should be zero. However, all centroids in the grey circle around A are closer to A than B. B is not even among the 20 nearest spatial neighbors of A! Thus distances among centroids do not capture neighborhood relationships well and give a distorted picture. Calculating the minimum distance between polylines A and B gives the correct result.

4.3 Distance measure

The distance is defined as the sum of the absolute distances among the (normalized) attributes

$$d(x_a, x_b) = \sum_{i=1}^{n} |x_{ai} - x_{bi}| .$$

We have fine-tuned the distance measure by assigning domain dependent weights to the attributes. However, we will not discuss the process here. The weights have been determined using background knowledge. The distance function is a major source to account for background knowledge. The minimum distance between spatially extended geographic objects is one component in the overall distance measure.

The prediction x_0 is the normalized weighted sum of the k nearest neighbors

$$x_0 = \frac{\sum_{i=1}^{k} w_i x_i}{\sum_{i=1}^{k} w_i},$$

where each weight w_i is a kernel $K(x_q, x_i)$ with

$$w_i = \frac{1}{d(x_0, x_i)} .$$

4.4. Partial evaluation

s-kNN uses as part of the overall (non-spatial) distance function the minimum distance between two polyline segments (or more generally, polygons). This is computationally more complex than calculating Euclidean distance between points. The distance calculation is done inside a spatial database that uses optimized code and indexing structures for spatial computations. Still, the overall running time of s-kNN is dominated by this comparison.

We utilize the general idea of *partial evaluation of the distance function* (e.g.[4]) and apply it to s-kNN. If a partial evaluation of the first *n* terms of the distance function shows that the instance is already farther away than some threshold – the distance of the current *k*-nearest neighbor – there is no need for further evaluation and we can avoid the expensive spatial calculation.

While iterating through the data set, the top *k* neighbors are stored in ascending order in a list and the threshold is adapted dynamically as instances are processed, being equal to the *k*-th nearest neighbor's distance (if there are no *k* neighbors in the list, the instance is inserted in the list).

If the value is smaller than the threshold, a spatial computation has to be performed. Here again we can save considerable computational effort if we apply a 2-step approach common to spatial databases [10]. For each polyline we calculate the Minimum Bounding Rectangle (MBR). The MBR for a set of points *P* is the minimum rectangular region with axis-parallel sides that encloses all of the points in *P* (fig. 3). Note, that the MBR for each object can be pre-computed and stored with the data. It is defined by a pair of x-y coordinates representing the main diagonal of the MBR. Calculating the minimum distance between two MBRs is much cheaper than calculating the distance between the actual polylines. Again, if the distance of the non-spatial attributes plus the distances between the MBRs is greater or equal to the threshold, the instance can be discarded.

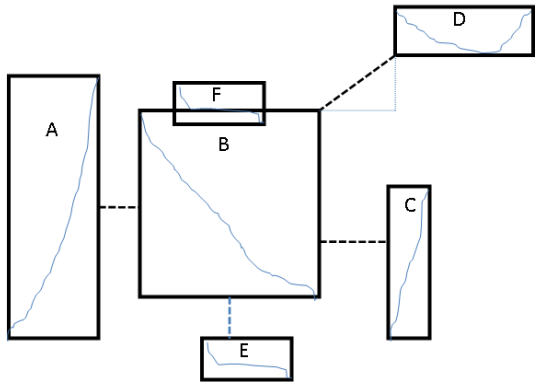

Figure 3. Minimum distances between Minimum bounding rectangles (dashed lines)

We show that the distance d_{min} between the polylines p_1 and p_2 is greater or equal to the distance of their MBR's $mbr(p_1)$ and $mbr(p_2)$.

Proof. (omitted due to space constraints) □

It should be stressed that s-kNN does not fall back to approximate solutions, but is *exact*. Partial evaluation is done for speed-up, but gives an exact solution for the nearest neighbors.

In the worst case, instances are evaluated in descending distance and we do not have any savings. In the best case, the best *n* instances are processed at the beginning; all others are farther away and need not to be fully evaluated. In practice, savings are somewhere in between, but very significant. For a city like Frankfurt a full computation would amount to 43 million spatial calculations (about 21,500 segments and 2,000 measurements). Using partial evaluation, calculations were sped up from nearly 1 day to about 2h (i.e. one order of magnitude). In addition, the dynamic calculations do not need to store distances. We also found empirically that the larger the problem, the greater the savings, because typically a roughly constant number of measurements in the surrounding have to be eva-

luated (typically several dozen), independently of the size of the overall data set.

The algorithm is implemented inside a spatial database (Oracle 10g). Oracle Spatial provides all necessary functionality for spatial processing. Using a database allows to process data that does not fit into main memory and allows for a tight integration between the data mining and pre-processing steps.

5. Experiments

For experiments and comparative evaluations of algorithms, we obtained a set of new measurements for the city of Rodgau, a 50,000 inhabitant city near Frankfurt. The set contains 51 measurements, which is sufficient for this size of city (fig. 4). Compared to the historic data set we are using for overall Germany, this data set has a number of advantages: a) measurements have been taken during the same period of a few days, b) the exact digital location of the measurements is known in each case and c) all categories of the street networks have been sufficiently covered.

However, Rodgau is not representative for larger, more complex structured cities. We therefore evaluated all algorithms additionally on the city of Hamburg, using a random sample of 2047 measurements, however with lower data quality. The target variable for both data sets is vehicular traffic.

The feature set was selected as described in Section 4.1. It includes vector geometries, the street category and type, and several buffer generated spatial densities of points of interests. For all standard non-spatial algorithms, we used simple centroid coordinates instead of vector features. We compared our algorithm to ordinary kNN, Model Trees, Gaussian processes, Support Vector Regression and Linear Regression. We used the algorithms as implemented in Weka [13] and extensively varied parameters for tuning. The best results for Model Trees were obtained with standard parameters. Gaussian Processes performed best with RBF kernels and reduced noise. For SVM Regression the best results were obtained for Rodgau with RBF kernels and for Hamburg with polynomial kernels of degree 3. Finally, Linear Regression achieved best performance using M5 attribute selection and elimination of colinear attributes.

We applied leave-one-out cross validation (only for Model Trees, Gaussian Processes and SVM Regression in Hamburg we used 10x cross validation due to excessive training times). As performance indicators, we calculated the correlation r and the relative absolute error (RAE) between predicted and true value. Table 1 shows the experimental results.

The s-kNN with 9 neighbors gave the best results, both in terms of relative absolute error (31.65% resp. 28.36%) and correlation (0.9503 resp. 0.9034). The 1-neighbor version performed not much worse, however, in cities with more outliers and possibly corrupted data, we found that 1-neighbor is far too unstable.

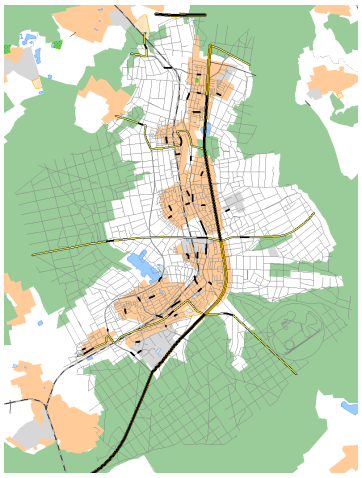

Figure 4. Measurement positions in Rodgau are indicted by thicker black lines

Table 1. Relative absolute error and correlation for Rodgau and Hamburg

	Rodgau		Hamburg	
	RAE	**Corr.**	**RAE**	**Corr.**
s-9NN	**31.65%**	**0.9503**	**28.36 %**	**0.9034**
s-5NN	32.51%	0.9494	28.70%	0.9022
s-1NN	32.50%	0.9500	29.98%	0.8946
9NN	43.77%	0.9310	32.09%	0.8967
5NN	39.81%	0.9389	31.35%	0.8937
1NN	48.42%	0.9018	30.19%	0.8799
Model Tree	40.99%	0.9023	48.86%	0.8209
Gauss. Proc.	40.41%	0.9283	48.62%	0.8293
SVM Regr.	42.93%	0.8921	47.70%	0.8151
Linear Regr.	44.40%	0.8928	56.20%	0.7690

On the Hamburg data set, algorithms outside the kNN family scored considerably lower. We believe that kNN has an advantage on this type of problem, as it can fit highly irregular structures. Gaussian processes work best if we can think of the phenomenon as a continuous field with some smooth surface, e.g. temperature. But clearly this is not how city traffic works.

In a second experiment, we evaluated the general importance of the spatial dimension. We removed the coordinates from the set of attributes and repeated the experiments. Table 2 shows the results for Hamburg along with the difference to the previous results.

Table 2: Results without spatial coordinates for Hamburg

	Hamburg without spatial attributes		Difference (w/o space – with space)	
	RAE	**Corr.**	**RAE**	**Corr.**
9NN	47.96	0.8190	15.88	-0.0777
5NN	47.89	0.8199	16.54	-0.0738
1NN	48.00	0.8160	17.81	-0.0639
Model Tree	50.75	0.8000	1.89	-0.0209
Gauss. Proc.	51.04	0.8053	2.42	-0.0240
SVM Regr.	49.93	0.7884	2.23	-0.0267
Linear Regr.	56.55	0.7639	0.35	-0.0051

All algorithms show a dependence on the spatial dimension. The dependence is especially strong in the kNN family, where RAE increases by more than 15 percent points if the space is left out. It shows the ability of kNN to directly exploit spatial autocorrelation and motivates its strength in the traffic domain. Note that even without the spatial dimension the kNN family obtains the best results. Finally, we note that doing the experiments with 10-fold cross-validation gives almost identical results as leave-one-out cross validation.

6. Summary

We described a data mining algorithm that infers a *frequency map*, where for each street segment in every German city larger than 50,000 inhabitants (approx. 1.3 Mio. segments), the model predicts the average number of cars and pedestrians per hour. We introduced s-kNN, a nearest neighbor based spatial data mining algorithm. It belongs to the class of vector-geometry based spatial data mining algorithms that reason on complex spatial objects instead. It does on the fly spatial computations based on a partial evaluation scheme. An empirical comparison with several state-of –the–art algorithms showed that s-kNN outperforms those methods both in terms of relative absolute error and correlation on this application.

The application has a strong impact on the way business is done in outdoor advertising: It has become the basis for pricing of posters for that branch of industry and is used for selecting new poster sites and planning of campaigns. By now, outdoor advertising perceives the project results as a backbone for their business.

7. Acknowledgements

The project has been supported by the German Fachverband Außenwerbung e.V. (FAW), with A. Schiefer and G. Schotten as responsible persons. Many partners contributed to data preparation, including GfK Germany, OSG, MGE Data and DDS. We gratefully acknowledge their contribution.

8. References

[1] Burrough, P., McDonnell, R. Principles of Geographical Information Systems, OUP, 1998

[2] Cressie, N. Statistics for Spatial Data, Wiley, 1993.

[3] Egenhofer, M. Reasoning about binary topological relations. In Gunther O. and Schek H.-J., (eds), In Second Symposium on Large Spatial Databases, volume 525 of LNCS, Springer, pages 143-160, 1991.

[4] Grother, P., Candela, G., Blue, J. Fast implementations of nearest neighbor classifiers, Pattern Recognition, 30(3): 459-465, 1997.

[5] Klösgen, W., May, M. Spatial Subgroup Mining Integrated in an Object-Relational Spatial Database. Proc. PKDD 2002, pages 275-283, 2002

[6] Koperski, K., Adhikary, J., Han, J. Spatial Data Mining: Progress and Challenges, SIGMOD'96 Workshop on Research Issues on Data Mining and Knowledge Discovery (DMKD'96), Montreal, Canada, 1996.

[7] Lisi, F.A, Malerba, D. Inducing Multi-Level Association Rules from Multiple Relations. Machine Learning, 55:175-210, 2004.

[8] Malerba, D., Appice, A., Cecci, M., Mining Model Trees from Spatial Data, Proc. PKDD 2005 LNCS, 2005.

[9] May, M., Scheider, S., Rösler, R., Schulz. D., Hecker, D. Pedestrian Flow Prediction in Extensive Road Networks using Biased Observational Data, ACM GIS 2008, to appear.

[10] Papadias, D., Sellis, T., Theodoridis, Y., Egenhofer. M. Topological relations in the world of minimum bounding rectangles: a study with R-trees, ACM Sigmod Record, 24,2: 92-103, 1995.

[11] Rinzivillo, S., Turini, F., Bogorny, V., Körner, C., Kuijpers, B., May, M., Knowledge Discovery from Geographical Data, in Giannotti, F., Pedreschi, D., Mobility, Data Mining and Privacy. Geographic Knowledge Discovery, Springer, pages 243-265, 2008.

[12] Sander, J. , M. Ester, H.-P. Kriegel, and X. Xu. Density-based clustering in spatial databases: The algorithm GDBScan and its applications. Data Mining and Knowledge Discovery, 2(2):169-194, 1998.

[13] Witten, I., Frank, E., Data Mining, Morgan Kaufman, 2005.