Improving Participatory Urban Infrastructure Monitoring through Spatio-Temporal Analytics

Matthias Budde, Julio De Melo Borges, Stefan Tomov, Till Riedel, Michael Beigl
Karlsruhe Institute of Technology (KIT), TECO / Pervasive Computing Systems, Karlsruhe, Germany
email: {budde, borges, tomov, riedel, michael}@teco.edu

ABSTRACT

Internet-enabled, location aware smart phones with sensor inputs have led to novel applications exploiting unprecedented high levels of citizen participation in dense metropolitan areas. Especially the possibility to make oneself heard on issues, such as broken traffic lights, potholes or garbage, has led to a high degree of participation in Urban Infrastructure Monitoring. However, citizens often repeatedly report duplicates instead of complementing information, leading to bottlenecks in manual processing by municipal authorities. Spatio-temporal clustering can serve as an essential tool to group and rank similar (e.g., duplicate) reports. Current data mining techniques could be used by municipal departments to manually process the most representative reports, but the mandatory parameter selection can be unintuitive, time consuming and error-prone. In this work, we therefore present a novel framework for clustering spatio-temporal data. We first apply an intuitive transformation of the data into a graph and subsequently use well-established graph clustering techniques to detect similar reports. We evaluate our method on two real-world data-sets from different mobile issue tracking platforms. As one of these datasets includes labels for duplicate reports, we can show how our framework outperforms existing techniques in our examplary use-case (duplicate detection).

Keywords

Crowdsourcing; Spatio-temporal Clustering; Duplicate Detection; Civic Issue Tracking; Issue Ranking;

1. INTRODUCTION

Participatory Sensing [3] has enabled a multitude of novel applications in recent years, ranging from collaborative noise pollution maps [29] over cyclist experience reports [23] to automatically characterizing places [4]. The crowdsourced gathering of content is becoming easier and faster as mobile and pervasive technology continues to spread. Instead of merely being a data collection paradigm, Participatory Sensing presents a powerful tool for harnessing civic engagement. Through geocentric crowdsourcing [7], administrative bodies and policy makers can utilize collaboratively collected data to gather information on various phenomena of interest, such as public infrastructure issues in urban spaces. Many different platforms for crowdsourced, mobile *Participatory Infrastructure Monitoring* or *Civic Issue Reporting* have emerged in the last years and enable citizens to make themselves heard and let them actively shape and connect to the urban spaces they live in: *FixMyStreet* [15] (UK), *SeeClickFix* [18], *CitySourced* (both in the US), *Fix-o-Gram* [10] (Australia), *Unortkataster Köln* [31], *KA-Feedback* [6] (both Germany) or *Nericell* [19] (India) are just some examples.

Citizens have a strong intrinsic motivation to enhance their living environment. As argued by FENNEL [9], crowdsourcing systems paradoxically may be negatively affected by a high degree of participation. Issue reporting will only be useful if the flow of records does not exceed the receiving entity's capacity to process or respond to reports. However, an increased data flow is often caused by submitted records being substitutes rather than complements:

- *The same* citizen may repeatedly report the same issue to emphasize the perceived urgency of an issue.
- *Different* users may see and report the same issue at different times and/or categorize the issue differently.

We confirmed that the frequent occurrence of duplicate reports is in fact an actual problem in the underlying real-word systems by gettin in contact with both the operator of a large platform, *SeeClickFix*[1], as well as the government partner of a small one: *KA-Feedback*[2], in the City of Karlsruhe, Germany. *SeeClickFix* currently does not have any mechanism that allows their government partners to handle similar reports in the system. However, they have emphasized the necessity of such a function to be implemented in the future. The municipal government of Karlsruhe also reported receiving multiple reports for unique infrastructure issues. A surprising revelation was that the current strategy for duplicate handling by case officers in Karlsruhe is keeping a mental record of the already processed reports. If this fails, they stated that the duplication is eventually detected by the maintenance crew that is sent out to fix the issue. Needless to say, this approach does not scale.

Employing intelligent (ideally real-time) data processing in general and spatio-temporal clustering in particular for grouping similar and possible duplicate reports has large potential:

- *Similar report detection and aggregation*: By automatically finding and grouping reports that are likely to describe the same unique issue, data processing entities can significantly speed up issue handling and better allocate their resources.

This paper is not considered an archival publication.
Copyright is held by the author/owner(s).
3rd ACM SIGKDD International Workshop on Urban Computing (UrbComp'14), August 24, 2014, New York, NY, USA

[1] http://www.seeclickfix.com
[2] http://www.ka-feedback.de

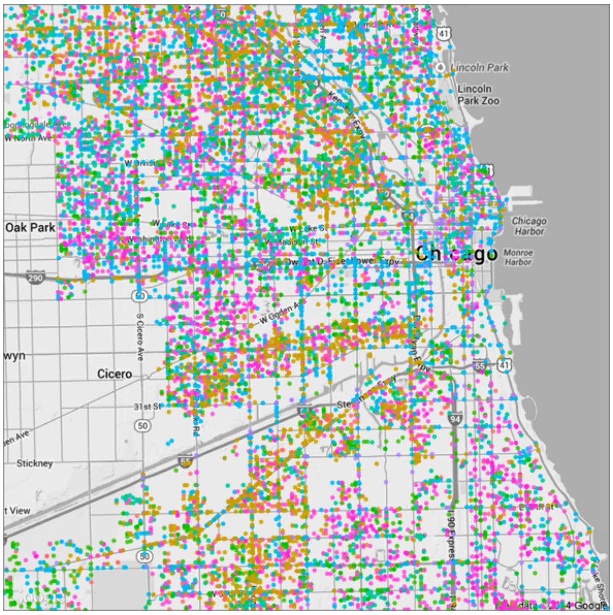

Figure 1: Spatial plot showing the SeeClickFix Chicago reports colored by distinct report category.

- *Implicit prioritization*: The size of the clusters discovered by data analytics can reveal infrastructure problems which are reported by many distinct users. This information can be exploited to prioritize the processing of certain specific issues.

In this paper, we present and evaluate a novel framework for clustering spatio-temporal data, based on initial transformation of the data into a graph structure and subsequent clustering of the graph. Figure 1 shows a spatial plot for a SeeClickFix dataset of the City of Chicago, illustrating the high spatial density of the reports. Performing clustering on the graph structure discards the need of setting unintuitive density parameters, as dense substructures can be directly discovered in the graph structure with traditional parameter free graph clustering approaches. We additionally compare the performance of our framework to contemporary spatio-temporal clustering methods and present a supervised approach for duplicate classification. It serves as a base for spatio-temporal knowledge discovery which can possibly help municipal government departments to easily and quickly rank and process groups of similar and even duplicate reports instead of each report individually.

2. RELATED WORK

In the field of spatio-temporal analytics, clustering methods have been distinguished by whether the analyzed data consists of *events* or *trajectories and moving points* [16]. In the Participatory Infrastructure Monitoring scenarios described above, we focus on analyzing reports, i.e., events. Spatio-temporal event analysis has been employed for a wide range of applications to better understand important aspects of urban phenomena. H. Silva et. al. have analyzed data from *Instagram* and *Foursquare* to analyze user's movement patterns and popular regions in a city [28, 27]. Hasan et. al. researched temporal and spatial aspects of the mobility and activity patterns in a city using twitter data [14]. However, to the best of our knowledge, applying spatio-temporal clustering for prioritizing the treatment of infrastructure reports by grouping similar and potentially duplicate reports is still an unexplored application.

Regarding current data mining techniques for spatio-temporal clustering, a well-established approach for spatial clustering is DBSCAN [8]. It clusters circular regions in which the density of an event is higher than outside within a certain (spatial) radius, but it only considers spatial events. Density based methods in general have various strong points over traditional spatial clustering like the traditional *k-means*, which requires knowing the number of clusters in the data a priori. DBSCAN needs no a priori assumption about the number of clusters, it can find arbitrarily shaped clusters and can perform well even in the presence of outliers. Based on DBSCAN, some algorithms have been proposed in order to separately identify spatial clusters and temporal clusters, and then combining the results in order to provide spatio-temporal clusters. Some examples are ST-DBSCAN [1] and ST-GRID [33]. ST-DBSCAN is an extension of DBSCAN with an additional threshold for the temporal dimension. ST-GRID is based on partitioning of the spatial and temporal dimensions into cells and defining dense cells as clusters. Adjacent dense cells get merged into a single cluster representation. Both algorithms suggest using the *k-dist-graph* proposed in [8] as a heuristic for determination of the spatio-temporal input parameters. They also have in common the requirement of a density threshold $minPts$, suggested to be set as $ln(n)$ on the number of data points in the database. However, the main problem of these density based approaches is that they cannot cluster data sets well with large differences in densities, since the combination of $minPts$ and the spatio-temporal thresholds cannot be chosen appropriately for all clusters [21]. Recent publications have addressed these drawbacks of density based clustering [17, 22]. These strategies however do not consider the temporal issue for the clustering.

Rather than relying on the definition of different density thresholds, we propose a novel approach based on modeling first the spatio-temporal data in form of a neighborhood connectivity graph. In a later step, dense structural subgraphs forming the spatio-temporal clusters are detected with parameter-free graph partitioning algorithms based on the structure of the modeled graph. We thus additionally overcome the known drawback of density based approaches of merging adjacent clusters which are connected by a very narrow dense link, as this is detected directly in the graph structure.

3. SPATIO-TEMPORAL ANALYTICS

In this work, we propose a generic approach that can be applied to the problem of grouping spatio-temporal close reports. We first describe the preliminary considerations that we undertook in the formation of our methods. Subsequently, the clustering framework is presented.

3.1 Preliminary Considerations

In order to develop our analyses, we define several terms and formal relations that we use to describe the spatio-temporal relationships. The data records we analyze are the notifications on infrastructure issues that residents of a city submit. These ***reports*** are by nature spatio-temporal observations, i.e. they have both a temporal and a spatial dimension (reporting time and location of the issue) and an issue description. Throughout this work we use the term ***user*** or *citizen* to describe a person who creates a report. The receiving entities that are responsible for processing, responding and reacting to the submitted reports are called ***case officers*** or *civil servants*.

Our main assumption is, for two reports to be similar and potentially describe the same infrastructure issue, they must be spatio-temporally neighbored. Two reports x and y ($x \neq y$) are considered to be ***spatio-temporal equivalent (ST-equivalent)*** or spatio-temporally neighbored, if for a temporal threshold (Δt) and a spatial distance (D) the following conditions are fulfilled:

$$|L(x) - L(y)| < D$$
$$|t(x) - t(y)| < \Delta t$$

The location of an observation x reported at time $t(x)$ is denoted as $L(x)$. Two reports x and y are therefore spatio-temporal equivalents, if for report x, submitted at time t, another report y is submitted within the timeframe Δt, and y lies within the spatial distance D from x. The spatio-temporal equivalence is by definition symmetric. If *x is a spatio-temporal equivalent of y*, reciprocally *y is a spatio-temporal equivalent of x*. In this sense, there is no notion of an original report. Naturally, there is a temporal order in which the reports are submitted, but it is not part of our model.

The spatio-temporal equivalence relation can be modeled and visualized as a graph, in which the vertexes represent the reports and which has an edge connecting two reports if they are spatio-temporally neighbored. We call this undirected and unconnected graph representation of the reports and their spatio-temporal relation the ***spatio-temporal graph (ST-Graph)***. In this graph, the spatial-temporal neighborhood is defined by linkage relationships between the objects.

On a set of reports R, we define the ***spatio-temporal neighbourhood*** $N_{st}(x) \subseteq R$ of a report x as the subset of reports from R that are spatio-temporal equivalents of x (not including x itself):

$$N_{st}(x) := \{y \in R \,|\, x, y \text{ are ST-equivalent}\}$$

From these two definitions it is clear that if for a report x to be *similar* to another report and even represent a possible duplicate, its spatio-temporal neighborhood is not empty and there must exist at least one edge connecting x to another report in the ST-Graph:

$$|N_{st}(x)| > 0 \Leftrightarrow Degree(x) > 0 \text{ in ST-Graph}$$

The spatio-temporal equivalence relation is not transitive, i.e. if x and y are ST-equivalent and y and z are ST-equivalent, that does not necessarily mean that x and z are as well. In order to reflect such an indirect connection between records, we introduce the notion of ***spatio-temporal connectivity***:

A report x is spatio-temporally connected (*ST-connected*) to another report y, if there is a chain of reports $r_0, ..., r_n$, such that $r_0 = x$ and $r_n = y$ and all r_i, r_{i+1} are pairwise spatio-temporal equivalent:

$$\begin{aligned} x, y \text{ ST-connected} :\Leftrightarrow \exists\, r_0, ..., r_n \,|\, r_i, r_{i+1} &\text{ ST-equivalent,} \\ r_0 = x, r_n = y&, \\ (\text{with } i \in \{0, ..., n-1\},\ n \in \mathbb{N}_{>0})& \end{aligned}$$

In the ST-Graph, if a path exists between two vertexes, these two are ST-connected.

Finally, we define a ***Spatio-Temporal Cluster*** C_{ST} as a non-empty subset of the set of reports R, where all reports in C_{ST} are spatio-temporally connected, fulfilling following condition (Maximality):

$$\forall x, y \in R : x \in C_{ST} \wedge x \text{ is ST-connected to } y \rightarrow y \in C_{ST}$$

The spatio-temporal clusters are connected components in the ST-Graph.

Using these preliminary considerations, the problem of finding and aggregating similar reports can now be reduced to the problem of constructing the ST-Graph with adequate parameters and performing graph clustering on its top.

3.2 Spatio-Temporal Clustering Framework

Using the spatio-temporal relationships introduced above, we developed a two-step framework for clustering spatio-temporal data (see Figure 2). In a first step (*Data Modeling*), we detect spatio-temporal equivalent reports given by spatial and temporal thresholds and some optional semantic constraints, forming the ST-Graph. In a second step (*Graph Clustering*), we apply graph clustering algorithms to detect and extract densely connected subgroups based on the structural similarity of the nodes in the ST-Graph. Spatio-temporal clusters are then extracted from the connected components of the graph and individual graph objects are defined as outliers. Some reports that were clustered in the first step may be removed from its cluster in this step.

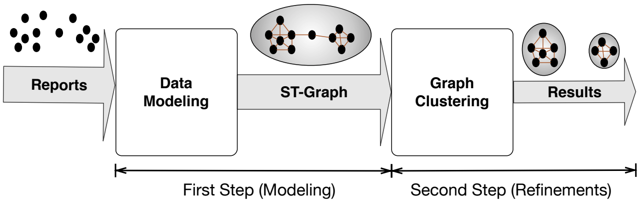

Figure 2: Two-step framework for clustering spatio-temporal data. In a first step, spatio-temporal proximity is modeled with the use of links between nodes. In a second refinement step, densely connected spatio-temporal clusters are partitioned.

3.2.1 Step 1: Data Modeling

The step of modeling the data into a graph structure is done by generating the *ST-Graph* with all reports as nodes linked to their respective spatio-temporal equivalent reports.

In the framework depicted in Algorithm 1, R represents the set of the reports to be analyzed and the $STGraph$ is the resulting unconnected and undirected graph after the data modeling. We introduce an optional parameter set $Constraints$ as input to the algorithm which allows specifying additional (non-spatio-temporal) conditions that should be included. This can contain arbitrary other relationships between reports, such as report category and user IDs of the submitting citizens. In the evaluation experiments of this work we have specified the additional constraint of building clusters only between reports of the same category (e.g. potholes, graffiti removal, etc.). In future work we aim to exploit text similarity metrics in the report descriptions instead, like the Leventhstein distance for example.

This first step of the framework is parameterized, i.e. the thresholds for the spatial and temporal distance (D, Δt) need to be predefined. The choice of these parameters can have a great influence on the clustering performance. The discussion section of this work contains considerations regarding how some of the constraints and the spatio-temporal parameters can influence the end result. The output of the first step of the framework may produce cluster representations consisting of subgraphs which are densely connected within themselves but sparsely connected in between. This phenomenon occurs more often in scenarios in which a high amount of (spatio-temporal dense) data is collected, and less frequent in cases where the data is spatio-temporal sparse. Consequently, further refinements steps become useful. In the following, we propose a refinement step that works on the base of the ST-Graph.

3.2.2 Step 2: Graph Clustering

The aim of the second step is to detect densely connected subgroups and reach a *partitioning of the clusters* in the ST-Graph, so that the resulting clusters are more *meaningful*. Densely connected subgraphs stand for clusters with high intra-cluster similarity. This is based on the assumption that clusters containing similar and potentially duplicate reports have highly dense connected vertexes and that dense subgraphs that are bridged by single connecting vertexes should be partitioned, i.e., connections within graph clusters should be dense, and the connections between different graph clusters should be sparse. The output (*ST-Graph*) of the first step of the algorithm already builds the clusters by retrieving the connected components (subgraphs) of the generated ST-Graph. The connected components of the ST-Graph would represent adequate clusters only if the clusters were distant from each other, but it is not satisfactory when clusters are near to each other. In some cases however, distinct clusters which are densely connected within themselves are merged over sparse connections in between. Nearby merged clusters can be detected and splitted-up on the basis of the graph and is illustrated in Figure 2.

```
Input: A set of reports $R$, temporal threshold $\Delta t$, spatial
       range $D$, and optionally an additional set of
       conditions $Constraints$
Output: Spatio-Temporal Clusters in $STgraph$

1  $STGraph \leftarrow$ new STGraph($R$)
2  $Clusters \leftarrow new$ List()
   /* First Step: data modeling                    */
3  foreach $i, j \in R : i \neq j$ do
4      if $\bigwedge_{CT \in Constraints} CT(i, j)$ then
5          if $|L(x) - L(y)| < D \wedge |t(x) - t(y)| < \Delta t$ then
6              $STGraph$.addEdge($i, j$)
7          end
8      end
9  end
   /* Second Step: graph clustering                 */
10 $CCS \leftarrow ConnectedComponents(STGraph)$
11 foreach $CC \in CCS$ do
12     $GPS \leftarrow GraphClustering(CC)$
13     foreach $GP \in GPS$ do
14         if size($GP$) $> 1$ then
15             $Clusters$.add($GP$)
16         end
17     end
18 end
19 return $Clusters$
```

Algorithm 1: Clustering framework that computes the *Spatio-Temporal Clusters* based on the generation of the *ST-Graph* and its subsequent partition.

Based on this premise, it is natural to partition the connected components of the ST-Graph, gathering vertices into groups such that there is a higher density of edges within groups than between them. This is done in line 12 of the framework. To combat this problem a number of new algorithms for graph clustering have been proposed upon different principles in recent years. Some are built around the idea of using centrality indices to find community boundaries, maximizing the number of intra-cluster edges optimizing the modularity measure [11]. It is not the focus of this work to exhaustively discuss which graph clustering algorithm would fit best in different cases. For the purpose if this work, we evaluated a modularity based algorithm which finds densely connected subgraphs in a graph by calculating the leading non-negative eigenvector of the modularity matrix of the graph [20]. This has proven to deliver good results in our evaluation which will be discussed in a later section. The result of applying the proposed clustering framework to the SCF Chicago dataset can be seen in Figure 3.

4. DATA COLLECTION

We collected data from two separate real-world issue tracking platforms that are actively used.

The first dataset originates from *KA-Feedback* (KAF), which represents an urban issue tracking platform in its beginnings. This means that the dataset which was collected over the first year of the platform's existence is still rather small (2,821 reports by February 2014) and relatively sparse, both spatially and temporally. The second dataset features a much larger amount of reports (34,690 entries altogether) which were also much denser compared to the first dataset. It contains open data that was extracted from the *SeeClick-Fix* [3] (SCF) platform. SCF is a large issue tracking platform that has deployments in several cities across the U.S., which also differ in size. The analyzed dataset is from one year of operation of the Chicago deployment of *SeeClickFix* (between February 2013 and 2014). In order to enable sustainable and comparable research results, we published this database as well as the clustering and additional evaluation results on our website [4] as benchmark for future publications. An exhaustive description of the dataset and its attributes can also be found on the website.

The *SeeClickFix* Chicago dataset has the unique feature that a large portion of the reports (32%) have been manually marked by Chicago municipal case officers as *duplicates*. However, it is noteworthy that the annotation of duplicate reports is not exhaustive. A coarse inspection of random samples from the data revealed, that there are in fact duplicate reports present, that have not been annotated as such. We exploited the manual labels for two purposes:

1. ***Evaluation and comparison of spatio-temporal clustering:*** Clusters are sets of objects in such a way that objects in the same group are in some sense *similar*. The highest degree of similarity is equality (*duplicates*). We thus expect the spatio-temporal clusters to precisely cover a high degree of duplicate reports. This will be exploited as evaluation metric in the evaluation section.

2. ***Assess supervised duplicate classification:*** Supervised machine learning models can be trained on the already manually classified data to learn to distinguish between duplicates and non-duplicate reports. This will be discussed in the next section.

5. SUPERVISED DUPLICATE CLASSIFICATION

The strength of the unsupervised framework for clustering similar and possible duplicate reports is strongly limited by the appropriate manual setting of the spatio-temporal parameters. Supervised machine learning models on the other hand possess the characteristic to optimally learn how other attributes a report, apart from the spatio-temporal ones, contribute to its duplicate status according to patterns determined by the data labels. In the following we will discuss the possibilities of assessing supervised machine learning for duplicate classification and how it can be combined

[3] `seeclickfix.com`, open data access under *Creative Commons Attribution-Noncommercial-Share Alike 3.0*.

[4] `http://www.teco.edu/~borges/urbcomp14/`

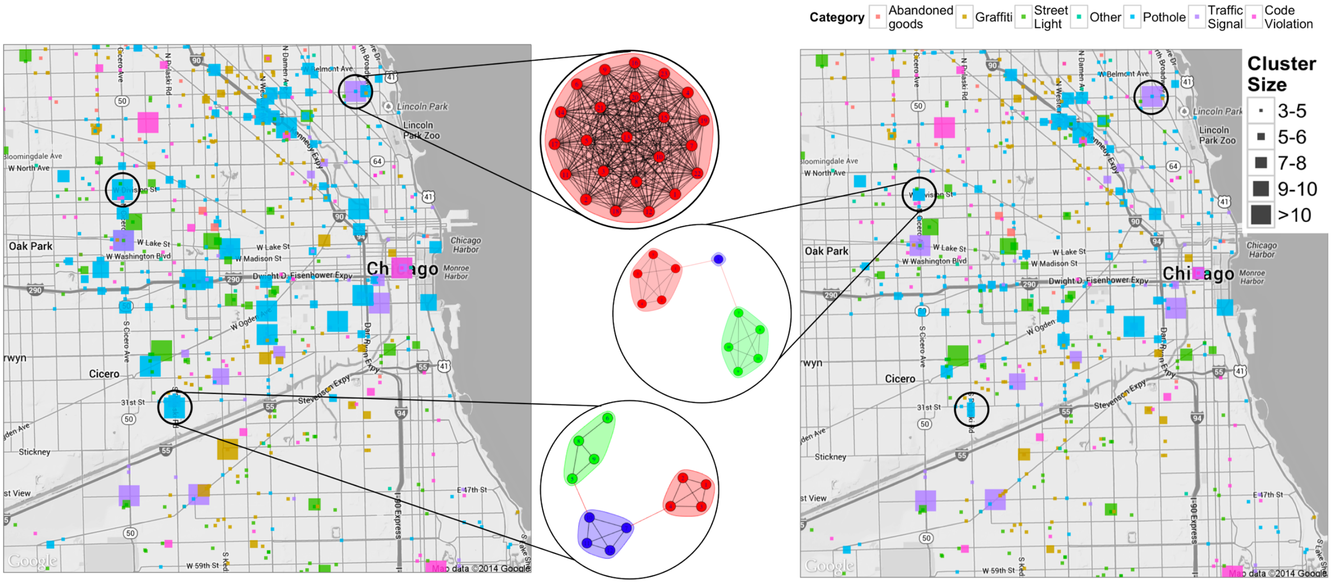

Figure 3: Results after the data modeling step (left) and the refinement step (right) of the proposed framework on the *SeeClickFix* Chicago dataset. Depicted are three graph representations of the clusters. One has been splitted into three distinct clusters after the refinement (bottom). Another has been splitted into two clusters, removing a report from its original cluster (middle).

with the unsupervised framework, contributing to creating a powerful duplicate classification model.

We exploited the duplicate labels in the *SeeClickFix* dataset in order to use the data for training a supervised predictor for duplicate classification. Some of the features have been directly extracted from the reports' description, like the *SecondsActiveUpdate*, which captures the temporal length the users interact with a report over the *SeeClickFix* portal website (time passed since the generation of the reports and its last update). Others were synthetically created based on the spatio-temporal neighborhood of the reports. An example are the features characterizing the spatial and temporal distance to the k-nearest neighbors of a given report (NN_i*-Spatial-Distance*, NN_i*-Temporal-Distance*). For our experiments we have considered the four (spatially) nearest neighbors, as we have not noticed any significant improvement by taking more neighbors into account. An exhaustive description of all features can be found in our website.

The model we used is a *Random-Forests* [2] with 500 trees. The choice for Random-Forests was based on its robustness against overfitting as well as on its natural capability to assess variable importance, used to evaluate the developed features. The importance of the feature vectors for the model has been measured by means of their role in decreasing the *Gini index* which the Random-Forest uses as node impurity measure. Figure 4 shows the results for the features we took into consideration and their suitability for duplicate detection, measured by their contribution to a decrease in the Gini Index.

As the most relevant features for the supervised classification (e.g. *Degree-in-ST-Graph* & *Is-in-Duplicate-Cluster*) are the exact same ones that we get as results from the unsupervised method, it stands to reason to combine the two approaches. Spatio-temporal patterns and other features that lead a report to be a duplicate can thus be recognized directly from the data, contributing to a more powerful enriched classification model. However, a fundamental difference between the two approaches is the fact that the supervised method delivers a classic binary classification result, whereas the clustering yields an aggregation of possible duplicates that belong together. Supervised methods that "*link*" duplicated data that belong together (pairwise duplicate classification) have already been investigated in the literature (cf. [25]) but have not been further investigated in this work due to nature of our datasets that lacks this kind of information.

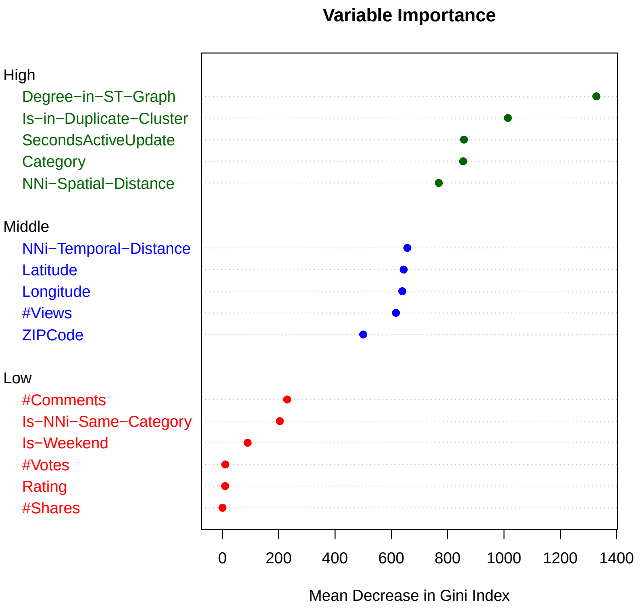

Figure 4: Set of features from the SCF dataset used for classifying duplicates. For each feature the importance for classification is compared by the mean decrease in the Gini Index for a Random Forest model.

6. EVALUATION

In the following, we will present the evaluation of our clustering framework and compare its performance to two contemporary density based spatio-temporal clustering algorithms: ST-GRID and ST-DBSCAN. For the purpose of evaluating the clustering and classification models we exploited the duplicate labels in the data. As the KAF dataset is unlabeled in terms of duplicates, a qualitative discussion of the results is given. For the SCF dataset labels are present and a detailed discussion of the quantitative evaluation follows.

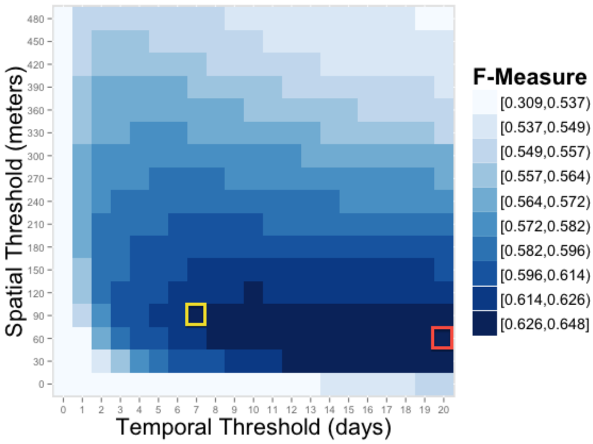

Figure 5: F-Measure Variation of our spatio-temporal clustering framework depending on spatio-temporal parameters.

The first evaluation regards the parametrization of the unsupervised framework, as it has great effect on the clustering qualities. In this specific evaluation case, we have tested the impact of the thresholds on both precision and recall on the first step of the algorithm regarding its ability to cluster duplicate reports: if the thresholds are too restrictive, the precision is high but the recall low. Are they too loose, the precision decreases and the recall increases. We have thus used the F-Measure, combining both precision and recall, to decide which parameters have the best impact on performance. This is presented in Figure 5.

In the heatmap, the overall best parameters are marked red ($D = 60m$ and $\Delta t = 20$ days, F-Measure $=$ **0.648**). The yellow mark represents the most restrictive parameters ($90m$ and 7 days, F-Measure $=$ **0.628**), that still deliver a good F-Measure. All our experiments in this section were run using these parameters, constraining the clustering only among reports of the same category (cf. line 4 of Algorithm 1).

In the following, we present the clustering results of running our spatio-temporal clustering on the *SeeClickFix* Chicago dataset (see Table 1). Our unsupervised approach groups 13,874 reports of the SCF Dataset, classified as similar and thus as possible duplicates by the algorithm, into 4,922 clusters. The KAF Dataset, containing 2,821 reports, revealed 285 clusters for 706 clustered reports. The graph partitioning step has had just a small effect on the KAF dataset compared to its effect on the SCF. Only 12 new clusters are generated after the partition. This is mainly due to the spatio-temporal density of the data. The KAF dataset is relatively sparse, both spatially and temporally containing only a few small clusters which are already densely connected, not being further partitioned in the second step of the framework.

Configuration	Dataset	SCF	KAF
	Parameters $D, \Delta t$	$90m$,$7d$	$90m$,$7d$
Before run	Total Reports	34,690	2,821
	Duplicates	11,121	*n/a*
After Step 1	Clustered reports	14,048	708
	Clusters	4,533	273
	Cluster size (μ, σ)	3.1, 2.46	2.59, 1.59
After Step 2	Clusters	4,922	285
	Cluster size (μ, σ)	2.85, 1.57	2.48, 1.05

Table 1: Results after running our spatio-temporal clustering on the *SeeClickFix* Chicago (SCF) and *KA-Feedback* (KAF) datasets.

Configuration	Dataset	SCF	KAF
	Parameters $D, \Delta t$	$90m$,$7d$	$90m$,$7d$
ST-GRID	Clustered reports	10,413	589
(mp=2)	Clusters	3,414	234
	Cluster size (μ, σ)	3.05, 2.48	2.51, 1.52
ST-DBSCAN	Clustered reports	16,947	981
(mp=2)	Clusters	5,147	375
	Cluster size (μ, σ)	3.29, 2.85	2.61, 1.47
ST-DBSCAN	Clustered reports	1,577	52
(mp=$\lfloor ln(n) \rfloor$)	Clusters	116	5
	Cluster size (μ, σ)	13.5, 4.21	10.4, 4.56

Table 2: Results after running ST-GRID and ST-DBSCAN on the *SeeClickFix* Chicago (SCF) and *KA-Feedback* (KAF) datasets with varying *minPts* (mp) parameter.

	ST	**ST**	**Clustering**	**Random**
	GRID	**DBSCAN**	**Framework**	**Forest**
F-Measure	**0.549**	**0.599**	**0.648**	**0.702**
Class.Error-True	0.468	0.243	0.231	0.339
Class.Error-False	0.190	0.362	0.285	0.104
Error Rate	0.28	0.32	0.27	0.18[5]

Table 3: Comparison of duplicate classification performance for the supervised and unsupervised models: F-measure, classification error for data labeled as duplicates (Error-True) respectively labeled as non-duplicate (Error-False) and the overall error rate.

The clustering algorithms ST-GRID [33] and ST-DBSCAN [1] were run on the same datasets using the same spatio-temporal parameters and its performance was compared to our approach. In addition to the spatial and temporal thresholds, both algorithms require a density threshold $minPts$. The proposed heuristic in [1] is to set this parameter to be $\lfloor ln(n) \rfloor$, n being the amount of reports in the dataset (see Table 2). To be able to better compare its results with our approach that does not discriminate dense regions based on a density threshold, we also tested it with $minPts = 2$ and evaluated its performance regarding duplicate detection. ST-DBSCAN has delivered a F-Measure of **0.599** and **0.187** and ST-GRID yields a F-Measure of **0.549** and **0.039** for $minPts = 2$ and $minPts = \lfloor ln(n) \rfloor$ respectively. Our approach thus outperforms both ST-GRID and ST-DBSCAN with respect to the given evaluation metrics.

The main problem of ST-GRID and in general of grid based methods regards its discretization: it is sensitive to grid position and resolution. For example, for a given density threshold τ, a

[5] Out-of-bag estimation of error rate (OOB)

cluster can be missed, even though it contains more than τ points in different cells. That's why it groups fewer reports and finds fewer clusters compared to other methods (cf. table 2). Density based methods like ST-DBSCAN or our clustering framework find clusters irrespective of orientation or size.

A comparison of the performance of the unsupervised models as well as the performance of the enriched supervised model (Random Forest) for the task of duplicate detection is given in Table 3. Overall, the enriched supervised model delivers lower error rates, outperforming all unsupervised models. However, labeled data is not always available. Additionally, in our case the labeling did not include information on which reports were "duplicate pairs", but this information would be especially interesting for better evaluating the different approaches.

7. DISCUSSION

Quantitatively, the performance of our presented methods points to a significant potential speed up of manual processing in the underlying application case. We believe that if case officers receive an aggregated view of the detected clusters, they could sort out similar reports and actual duplicates much more quickly and link back to the original report. However, the prove that a semi-automatic system using the proposed framework outperforms a manual system when also looking at human factors has yet to be made. Aside from that, the following subsections discuss some technical limitations.

The two methods we presented each have their specific strengths and drawbacks. Their suitability strongly depends on the application case.

7.1 Spatio-Temporal Feature Selection

The output of the clustering framework and the spatio-temporal features we created are the features with the most predictive power for the supervised duplicate classification as illustrated in Figure 4. Once more, this confirms the strong relation between the spatio-temporal neighborhood of a report and its likelihood to be a duplicate. Interestingly, to date only a variety of features ranging from text comparison to numeric similarities have been exploited for supervised duplicate detection [5, 32]. However, spatio-temporal relationships between records have not been exploited. Our research thus provides new insights regarding the similarity metrics for relationships of records in general, and in the use of spatio-temporal attributes as an input feature for a supervised classification in particular.

7.2 Category-Sensitive Parametrization

Regarding the appropriate selection of the spatial and temporal thresholds for our clustering framework, we evaluated harnessing additional meta-information of the reports. One possibility we explored is to set the temporal thresholds according to the mean time it takes to the municipal government to fix an issue (*mean-time-to-fix* (*MTF*). In total 16.34% of the reports in the *SeeClickFix* Chicago dataset are marked as fixed, wherein the overall MTF is 13.47 days ($\sigma = 30.56$). The threshold can also be set differently (or even dynamically) for different categories of reports, as we observed that the MTF strongly differs across categories (see Figure 6). For example, broken traffic lights tend to get fixed within a day or two, while potholes or abandoned goods are generally tended to later. The means are depicted as white dots on the boxes. To enable better visualization the plot as been cut on the $30th$ day, obstructing the visualization of the box for the category *Code Violations*, which has a mean and a third quantile range from 78.09 and 190.0 days respectively. These categories have been artificially created through text-mining based on the description of the report.

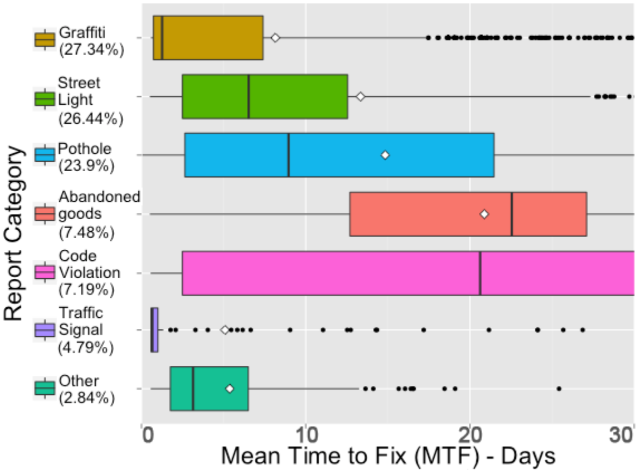

Figure 6: Mean Time-To-Fix (MTF) for different issue categories, sorted by frequency of occurrence.

The category "Code Violations" for example refers to reported issues about Building or Sanitation Code Violation.

For the duplicate evaluation, applying dynamic parameters has led to a best F-Measure of 0.638 tested with varying spatial parameters. Dynamic category-sensitive parametrization has thus not delivered any measurable advantages over the global best parameters, which yields a F-Measure of 0.648.

7.3 Deriving Issue Importance from Cluster Size

We found the great majority of the clusters to be dyads after the first step of the framework on the SCF dataset ($\mu = 3.107$, $Q_2 = 2.0$, $Q_3 = 3.0$). 21.1% of the clusters are of size 4 or higher, reaching a maximum of size 34. The size of the clusters can deliver valuable information on the urgency of an infrastructure issue, which can help civic servants not only to speed-up manual issue handling by processing multiple issues at once but also to potentially prioritize certain issues and better allocate their resources. As by mean of example we found a relatively big cluster in the KAF dataset containing descriptions of broken glass fragments on a cycleway. Such big clusters are prone to describe issues which affected a high number of citizens and can be easily discovered using spatio-temporal clustering.

8. CONCLUSION

In this work, we have presented two spatio-temporal data analysis techniques and evaluated them on the data of two separate real-world issue tracking platforms, one in its beginnings (2,821 reports) and one mature (close to 35,000 reports). The first method, an unsupervised clustering framework, uses parametrized spatio-temporal clustering requiring only a temporal and spatial threshold as parameters and providing very promising results for particular applications. We have evaluated and compared our approach to the state-of-the-art spatio-temporal clustering algorithms ST-Grid and ST-DBSCAN, showing that our approach does even outperform both approaches, delivering results which give excellent agreement with expected results. The second method is a supervised machine learning approach using spatio-temporal features and Random Forests for duplicate classification. Our results show that we can automatically detect duplicate infrastructure reports with reasonably low error rates (18%).

This work presents an innovative use-case application of data mining techniques to urban infrastructure monitoring. While with growing participation, we see immediate need for such technologies in clustering and detecting possible duplicate reports, smaller applications deal with other challenges, such as the reluctance of a (mostly older) user base to go digital or the surprising efficiency of manual operation. One result we got from interviewing was, however, that clustering could help humans better understand grouping report information, so that semi-automated technologies are worth investigating.

We have made the evaluated dataset publicly available and published exhaustive information about our evaluation on our website, in order to enable other researchers to compare their findings and use our results as a benchmark for future work.

9. FUTURE WORK

While the framework presented in this paper can already greatly facilitate manual processing of records from civic issue reporting platforms, we intend to investigate how the application case could benefit from further refinements and/or combination with other techniques. Specifically, we aim to address three challenges in future work.

Open Challenge 1: Online processing

In this work, we apply our framework offline. However, participatory urban infrastructure monitoring would greatly benefit from online and real-time processing of the data. This would allow the detection of similar reports at the time new data is submitted by a user. Due to the flexibility of our proposed framework regarding the clustering, it is in principle well suited to work online and in real-time. There are already several graph clustering techniques for online clustering [13]. The insertion or deletion of an object in a cluster affects the current clustering only in the spatio-temporal neighborhood of this object. Every time a report is submitted, the algorithm can search its spatio-temporal neighborhood in real-time for similar reports, updating the graph structure model. If any is found, a new cluster can be allocated or the report is added to an already existing cluster. Nevertheless, the main challenge thereby is to model and evaluate an accurate data transformation for the framework in order to enable this online preprocessing in the first step of our framework.

Open Challenge 2: Enriching the model with more information

The clustering framework presented only considers spatio temporal information, but we can extract more information from the reports in order to achieve more accurate results. For example, infrastructure issue reports constitute spatio-temporal observations with semantic content. In our experiments, we have shown this with an supervised method for duplicate detection that considers other attributes from the reports. However, we aim to introduce this information within the graph model for more accurate results. In particular, we want to focus on the semantic hidden in the user reports. In this work, we have trivially leveraged semantic similarity by narrowing the spatio-temporal neighborhood to be only applicable to reports of the same category. In future work we aim to combine text analysis (e.g. on issue description) with spatio-temporal features of the data. To achieve this, we can also apply existing graph clustering model for attributed graphs [34, 12]. In the so called *attributed graphs* each object is represented by its relationships to other objects (edge structure) and its individual properties (node attributes) [24]. Using this in our use-case, the ST-Graph can also contain attribute information of the data. This would allow us to find better patterns in the data that does not only relies on the modeled graph structure but also embraces a set of other attributes besides the spatio-temporal ones. Additionally, we can also enrich our ST-Graph with different types of nodes (e.g., users as nodes) or edges (e.g., friendships between users) using clustering techniques for heterogeneous networks [30]. Although we can apply such novel clustering algorithms for different graph types within our framework, it is essential to provide our model with an accurate data transformation of the initial reports provided by the users in order to obtain good results.

Open Challenge 3: Scalability / Transferability

To the best of our knowledge, there are currently no clustering or duplicate detection technologies being employed in large scale participatory sensing applications. Therefore we would like to test these methods in the actual operation of such large applications in the near future. This is not limited to the application case of urban infrastructure monitoring. As the approach presented in this paper works on spatio-temporal information, it can also be applied to many other applications. This would allow us to generalize the approach and evaluate it on other big spatio-temporal datasets. As social networks like Twitter continue to increase the use of geotagging, social networks are being increasingly researched to better understand city dynamics and urban social behavior, delivering location and time aware information which is an important aspect of the urban phenomena [28]. Therefore, as we are researching enrichments of our model with more information, this will open up new possibilities to be considered in the future. On the other hand, only using a temporal and a spatial parameter makes our framework applicable to use cases, in which data has no additional semantic information. One scenario which can greatly benefit from spatio-temporal analysis is the calibration of distributed low-cost sensors [26].

Acknowledgments

This work was partially funded by the German Federal Ministry of Education and Research (BMBF) as part of the ESTAData Project (grant number 01IS12051). The authors would like to thank PATRICIA IGLESIAS SÁNCHEZ of the *Institute for Program Logic and Data Structures (IPD)* at KIT for most stimulating discussions and valuable input while reviewing this work. We further thank the *Research Center for Information Technology* (FZI) in Karlsruhe and the *Medienbüro* of the City of Karlsruhe for the kind permission to analyze their *KA-Feedback* dataset, as well as the participation in our structured interview. Many thanks also go to `seeclickfix.com`, for providing open access to their datasets (under *CC Attribution-Noncommercial-Share Alike 3.0*) and permitting us to analyze them for our research, as well as kindly providing insights into their current strategies and needs.

References

[1] D. Birant and A. Kut. St-dbscan: An algorithm for clustering spatial–temporal data. *Data & Knowledge Engineering*, 60(1):208–221, 2007.

[2] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[3] J. Burke, D. Estrin, M. Hansen, A. Parker, N. Ramanathan, S. Reddy, and M. B. Srivastava. Participatory Sensing. In *Workshop on World-Sensor-Web (WSW'06): Mobile Device Centric Sensor Networks and Applications*, pages 117–134, 2006.

[4] Y. Chon, N. D. Lane, F. Li, H. Cha, and F. Zhao. Automatically characterizing places with opportunistic crowdsensing using smartphones. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, UbiComp '12, pages 481–490, New York, NY, USA, 2012. ACM.

[5] M. Cochinwala, V. Kurien, G. Lalk, and D. Shasha. Efficient data reconciliation. *Information Sciences*, 137(1-4):1 – 15, 2001.

[6] J. De Melo Borges, V. Zacharias, and N. Plessing. PartSense: A Participatory Sensing Platform and its Instantiation KA-Feedback. In *Geoinformatik 2012*, 2012.

[7] T. Erickson. Geocentric Crowdsourcing and Smarter Cities: Enabling Urban Intelligence in Cities and Regions. In *1st Ubiquitous Crowdsourcing Workshop at UbiComp*, 2010.

[8] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD*, volume 96, pages 226–231, 1996.

[9] L. Fennell. Crowdsourcing land use. *Brooklyn Law Review*, 78, 2013.

[10] M. Foth, R. Schroeter, and I. Anastasiu. Fixing the city one photo at a time: Mobile logging of maintenance requests. In *23rd Australian Computer-Human Interaction Conference*, OzCHI '11. ACM, 2011.

[11] M. Girvan and M. E. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826, 2002.

[12] S. Gunnemann, I. Farber, B. Boden, and T. Seidl. Subspace clustering meets dense subgraph mining: A synthesis of two paradigms. In *Data Mining (ICDM), 2010 IEEE 10th International Conference on*, pages 845–850. IEEE, 2010.

[13] T. Hartmann, A. Kappes, and D. Wagner. Clustering evolving networks. *CoRR*, abs/1401.3516, 2014.

[14] S. Hasan, X. Zhan, and S. V. Ukkusuri. Understanding urban human activity and mobility patterns using large-scale location-based data from online social media. In *Proceedings of the 2nd ACM SIGKDD International Workshop on Urban Computing*, page 6. ACM, 2013.

[15] S. F. King and P. Brown. Fix my Street or Else: Using the Internet to Voice Local Public Service Concerns. In *1st International Conference on Theory and Practice of Electronic Governance*, pages 72–80. ACM, 2007.

[16] S. Kisilevich, F. Mansmann, M. Nanni, and S. Rinzivillo. Spatio-temporal clustering. In O. Maimon and L. Rokach, editors, *Data Mining and Knowledge Discovery Handbook*, pages 855–874. Springer US, 2010.

[17] P. Liu, D. Zhou, and N. Wu. Vdbscan: varied density based spatial clustering of applications with noise. In *Service Systems and Service Management, 2007 International Conference on*, pages 1–4. IEEE, 2007.

[18] I. Mergel. Distributed Democracy: SeeClickFix.com for Crowdsourced Issue Reporting. *Social Science Research Network*, 2012.

[19] P. Mohan, V. N. Padmanabhan, and R. Ramjee. Nericell: rich monitoring of road and traffic conditions using mobile smartphones. In *Proceedings of the 6th ACM conference on Embedded network sensor systems*, pages 323–336. ACM, 2008.

[20] M. E. Newman. Finding community structure in networks using the eigenvectors of matrices. *Physical review E*, 74(3):036104, 2006.

[21] M. Parimala, D. Lopez, and N. Senthilkumar. A survey on density based clustering algorithms for mining large spatial databases. *International Journal of Advanced Science and Technology*, 31(1), 2011.

[22] A. Ram, S. Jalal, A. S. Jalal, and M. Kumar. A density based algorithm for discovering density varied clusters in large spatial databases. *International Journal of Computer Applications*, 3(6):1–4, 2010.

[23] S. Reddy, K. Shilton, G. Denisov, C. Cenizal, D. Estrin, and M. Srivastava. Biketastic: Sensing and mapping for better biking. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pages 1817–1820, New York, NY, USA, 2010. ACM.

[24] P. I. Sánchez, E. Müller, F. Laforet, F. Keller, and K. Böhm. Statistical selection of congruent subspaces for mining attributed graphs. In *International Conference on Data Mining (ICDM), Proceedings of the 13th IEEE*. IEEE, 2013.

[25] S. Sarawagi and A. Bhamidipaty. Interactive deduplication using active learning. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 269–278. ACM, 2002.

[26] O. Saukh, D. Hasenfratz, C. Walser, and L. Thiele. On rendezvous in mobile sensing networks. In K. Langendoen, W. Hu, F. Ferrari, M. Zimmerling, and L. Mottola, editors, *Real-World Wireless Sensor Networks*, volume 281 of *Lecture Notes in Electrical Engineering*, pages 29–42. Springer International Publishing, 2014.

[27] T. H. Silva, P. O. Melo, J. M. Almeida, J. Salles, and A. A. Loureiro. Visualizing the invisible image of cities. In *Green Computing and Communications (GreenCom), 2012 IEEE International Conference on*, pages 382–389. IEEE, 2012.

[28] T. H. Silva, P. O. Vaz de Melo, J. M. Almeida, J. Salles, and A. A. Loureiro. A comparison of foursquare and instagram to the study of city dynamics and urban social behavior. In *Proceedings of the 2nd ACM SIGKDD International Workshop on Urban Computing*, page 4. ACM, 2013.

[29] M. Stevens and E. D'Hondt. Crowdsourcing of pollution data using smartphones. In *Workshop on Ubiquitous Crowdsourcing at Ubicomp'10*, 2010.

[30] Y. Sun and J. Han. Mining heterogeneous information networks: Principles and methodologies. *Synthesis Lectures on Data Mining and Knowledge Discovery*, 3(2):1–159, 2012.

[31] G. Trogemann, S. Göllner, and L. Scherffig. Unortkataster: An Urban Experiment Towards Participatory Media Development. 2008.

[32] V. S. Verykios, A. K. Elmagarmid, and E. N. Houstis. Automating the approximate record-matching process. *Information Sciences*, 126(1-4):83 – 98, 2000.

[33] M. Wang, A. Wang, and A. Li. Mining spatial-temporal clusters from geo-databases. In *Advanced Data Mining and Applications*, pages 263–270. Springer, 2006.

[34] Y. Zhou, H. Cheng, and J. X. Yu. Graph clustering based on structural/attribute similarities. *Proceedings of the VLDB Endowment*, 2(1):718–729, 2009.