Effectively Modeling Data from Large-area Community Sensor Networks

Saket Sathe§  Sebastian Cartier§  Dipanjan Chakraborty‡  Karl Aberer§
§EPFL, Switzerland. {name.surname}@epfl.ch
‡IBM Research India, cdipanjan@in.ibm.com

ABSTRACT

Effectively managing the data generated by Large-area Community driven Sensor Networks (LCSNs) is a new and challenging problem. One important step for managing and querying such sensor network data is to create abstractions of the data in the form of models. These models can then be stored, retrieved, and queried, as required. In our OpenSense1 project, we advocate an adaptive model-cover driven strategy towards effectively managing such data. Our strategy is designed considering the fundamental principles of LCSNs.

We describe an adaptive approach, called adaptive k-means, and report preliminary results on how it compares with the traditional grid-based approach towards modeling LCSN data. We find that our approach performs better to model the sensed phenomenon in spatial and temporal dimensions. Our results are based on two real datasets.

Categories and Subject Descriptors
I.5.3 [Clustering]: Algorithms

General Terms
Algorithms, Design, Performance

Keywords
adaptive clustering, community sensing, data management

1. INTRODUCTION

OpenSense1 is a LCSN whose major scientific objective is to investigate challenges in efficiently and effectively monitoring environmental parameters (e.g., air pollution) using community-driven sensors, mounted on buses and cars. In this context, our work investigates different approaches of synthesizing the data generated by LCSNs. At its core, LCSNs form a dynamic new form of mobile geosensor networks, characterized by uncontrolled or semi-controlled mobility of vehicles or people, moving over a large geographical area. For this reason, we treat the underlying sensor network as a disconnected component, which is collecting data using local policies and principles.

Although, there is significant literature on model-based query processing, both in-network [1] or in the back-end [3], on mobile sensor networks, there is a lack of understanding of approaches to determine high quality and concise models of the phenomenon from LCSNs. Most prior work [4, 5] implicitly assumes that the sensors are relatively homogeneously distributed and/or their sensing behavior can be tuned, considering the phenomenon being sensed.

Unfortunately, this is not true for LCSNs. Hence, it is difficult to produce a homogeneous, good quality view of the phenomenon. The community-sensing pattern often leads to spatio-temporal irregularities in sensing. Therefore, a challenging question is: how do we efficiently create quality-controlled models that cover the sensed data, spatially and temporally?

Towards this, we propose adaptive strategies that discover spatial areas that can be modeled using single or multiple models. Our strategies adapt to the changing nature of the sensed phenomenon by adjusting the geographical granularity of the models to capture the phenomena with high fidelity. Through user-defined approximation error thresholds, we determine the quality of the models demanded.

![Figure 1: Architecture of the framework.](image)

We compare our approach with a grid-based model cover strategy called GRIB, where the area under consideration \( \mathcal{R} \) (refer Figure 1) is divided into equal size grid cells and a model is estimated for each grid cell. We find that adaptive approaches are significantly better with respect to the tradeoff between computational complexity and model quality, and also towards modeling the temporal evolution of the phenomenon.

2. ADAPTIVE MODELS

Our adaptive modeling approach provides a multi-model abstraction or a model cover over the raw tuples dumped in the region \( \mathcal{R} \) (refer Figure 1). A model cover is a set of models \( \mathcal{M} = \{M_1, \ldots, M_p\} \) that are respectively responsible for modeling the sub-regions \( \mathcal{R}_1, \mathcal{R}_2, \ldots, \mathcal{R}_p \) of \( \mathcal{R} \). The sub-regions, taken together, cover the entire region \( \mathcal{R} \). Specifically, the model cover is responsible for two tasks: (i) estimate the models \( M_1, M_2, \ldots, M_p \), such that the approximation error per model and the total number of models (p) are minimized, and (ii) efficiently maintain the model cover as and when there are changes to the observed phenomena.

**Estimating the Model Cover:** For our first task, we present an adaptive method, called adaptive k-means or Ad-KMN, that gave us the best results amongst many candidates we designed [2]. This method partitions the region \( \mathcal{R} \) adaptively (i.e., only when and where it is necessary) and estimates the models \( M_1, M_2, \ldots, M_p \). The standard k-means algorithm uses the Euclidean distance for creating the clusters. Instead, in the Ad-KMN method, we use the model approximation error as an additional clustering criteria.
We denote the raw tuple as \( b_i = (t_i, x_i, y_i, r_i) \), where \( r_i \) is the raw sensed value, and \( t_i \) and \( (x_i, y_i) \) are time and position corresponding to the sensed value \( r_i \). We assume that the model cover is computed using a window of raw tuples \( W_s \). \( W_s \) is a set of raw tuples \( b_i \), whose time \( t_i \) is in between \( sH \) and \((s+1)H\), where \( s \) is a positive integer and \( H \) is the window length.

An example of the Ad-KMN method on toy data is shown in Figure 2. Assume that before executing the Ad-KMN method, we compute two centroids \( \mu_1 \) and \( \mu_2 \) by executing the standard k-means algorithm using the positions \( (x_i, y_i) \) found in the raw values of the window \( W_s \) (refer Figure 2(a)). We, then, (a) partition the raw values in \( W_s \), such that \( R_1 \) and \( R_2 \) contain raw values that are nearest to \( \mu_1 \) and \( \mu_2 \) respectively, and (b) for the raw values in \( R_1 \) and \( R_2 \) we estimate linear regression models \( M_1 \) and \( M_2 \) and compute the approximation error.\(^2\)

Next, we check whether the approximation error is within a user-defined threshold \( \tau_n \). If, for instance, the error in both the regions \( R_1 \) and \( R_2 \) greater than \( \tau_n \), then we introduce an additional cluster centroid for each region \( R_1 \) and \( R_2 \) and re-estimate the four centroids \( \mu_1, \mu_2, \mu_3, \mu_4 \) using the standard k-means algorithm (refer Figure 2(b)). This procedure is continued until all regions meet the approximation error threshold \( \tau_n \).

Figure 2: Example on toy data: (a) initial regions, and (b) two new regions \( R_3 \) and \( R_4 \) added after an Ad-KMN iteration.

**Efficiently Maintaining the Model Cover.** Furthermore, we are interested in maintaining the model cover as new windows of raw tuples are streamed into the system. Given several windows of raw tuples \( W_s \), where \( s = (1, 2, \ldots, S) \), we are interested in continuously maintaining the model cover, while reducing the number of additional computations required for its maintenance.

We start by estimating the cluster centroids \( \mu_1, \mu_2, \ldots, \mu_p \) over a training window \( W_D \) of size \( D \gg H \) using the Ad-KMN method. The Ad-KMN method returns the regions \( R_o \) and models \( M_o \), where \( \alpha = (1, \ldots, p) \). Now, assume that the first window of raw values \( W_1 \) is available. \( W_1 \) is first partitioned according to the cluster centers \( \mu_\alpha \), such that \( W_1^o \) contains the raw tuples nearest to \( \mu_\alpha \). Next, if the approximation error obtained using the raw values in the partition \( W_1^o \) is greater than a user-defined model retain threshold \( \tau_o \), then we invalidate the model \( M_o \) and re-estimate it from scratch. We perform a similar test for all the other \( W_s \). We use flops\(^3\) to measure the re-estimation cost of the model cover.

3. **PRELIMINARY EXPERIMENTS**

We demonstrate results obtained using our Ad-KMN method on two real datasets collected from large geographical areas. The opensense dataset contains 110k raw tuples measuring the Ozone (O\(_3\)) concentration in Zurich, Switzerland over a period of seven weeks, through bus-mounted sensors. The safecast\(^4\) dataset contains 979k raw tuples of radiation values collected by the community in Eastern Japan after the Fukushima Daiichi nuclear disaster.

**Error Analysis.** Figure 3 shows the approximation error as a function of the number of regions \( p \), where \( \tau_n \) is set to 1% and \( H \) is

\(^{2}\text{approximation error} \) is the average percentage error compared to the normal range of \( r_i \) in the environment (pollutant specific).

\(^{3}\)A flop represents either an addition or a multiplication of two floating point numbers.

\(^{4}\)http://blog.safecast.org/

4. **CONCLUSIONS**

In this paper, we demonstrated that adaptive model cover estimation methods, namely Ad-KMN, exhibit promising performance gains in terms of accuracy and efficiency as compared to the grid-based methods for modeling obtained from LCSNs. Future work will focus on efficient storage and querying processing using adaptive model covers.

5. **REFERENCES**


