A Cloud-Based Knowledge Discovery System for Monitoring Fine-Grained Air Quality

Yu Zheng[1], Xuxu Chen[1,2], Qiwei Jin[1], Yubiao Chen[1,3], Xiangyun Qu[1], Xin Liu[1], Eric Chang[1], Wei-Ying Ma[1], Yong Rui[1], Weiwei Sun[2]

[1]Microsoft Research, No.5 Danling street, Haidian District,Beijing 100080, China

[2]Fudan University, Shanghai, China

[3]Harbin Institute of Technology, Harbin, China

{yuzheng, v-xuche, qiwjin, v-yubche, xinliu, echang, wyma, yongrui}@microsoft.com, wwsun@fudan.edu.cn

ABSTRACT

Many developing countries are suffering from air pollution recently. Governments have built a few air quality monitoring stations in cities to inform people the concentration of air pollutants. Unfortunately, urban air quality is highly skewed in a city, depending on multiple complex factors, such as the meteorology, traffic volume, and land uses. Building more monitoring stations is very costly in terms of money, land uses, and human resources. As a result, people do not really know the fine-grained air quality of a location without a monitoring station. In this paper, we introduce a cloud-based knowledge discovery system that infers the real-time and fine-grained air quality information throughout a city based on the (historical and real-time) air quality data reported by existing monitor stations and a variety of data sources observed in the city, such as meteorology, traffic flow, human mobility, structure of road networks, and point of interests (POIs). The system also provides a mobile client, with which a user can monitor the air quality of multiple locations in a city (e.g. the current location, home and work places), and a web service that allows other applications to call the air quality of any location. The system has been evaluated based on the real data from 9 cities in China, including Beijing, Shanghai, Guanzhou, and Shenzhen, etc. The system is running on Microsoft Azure and the mobile client is publicly available in Window Phone App Store, entitled Urban Air. Our system gives a cost-efficient example for enabling a knowledge discovery prototype involving big data on the cloud.

Categories and Subject Descriptors

H.2.8 [Database Management]: Database Applications - data mining, Spatial databases and GIS;

General Terms

Algorithms, Management, Experimentation

Keywords

Urban computing, air quality, city dynamics, human mobility.

1. INTRODUCTION

Many developing countries, e.g., China, Brazil, and India, are suffering from air pollution recently. Many governments have built air quality monitoring stations in cities to inform people the real-time concentration of air pollutants, such as $PM_{2.5}$. In reality, however, a city has insufficient air quality monitoring stations because building and maintaining such a station is very costly in terms of money, land uses, and human resources [6]. Unfortunately, urban air quality varies by locations significantly and is highly skewed in a city, as it depends on multiple complex factors, such as meteorology, traffic, land use, and urban structures. For instance, Beijing has 22 stations in the urban spaces, as depicted in Figure 1 A). However, according to the statistics on the air quality index (AQI) recorded from Jan. 1, 2013 to Jan. 1 2014, the average deviation between the maximum and minimum readings of $PM_{2.5}$ from the 22 stations at the same timestamp can easily exceed 120, as shown in Figure 1 B). In addition, over 50% of time, the deviation is larger than 100, as depicted in Figure 1 C). 100 almost denotes a two-level gap, i.e., when the air quality of a location is moderate, another one could be unhealthy.

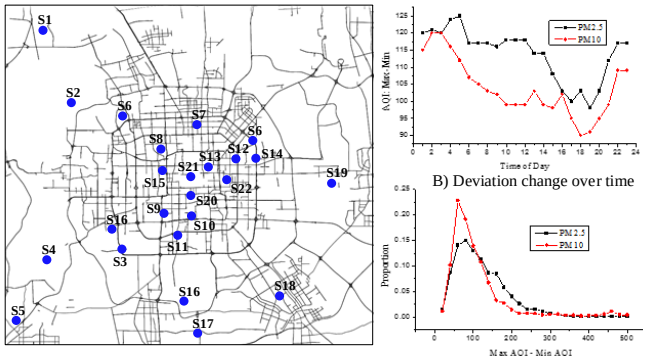

Figure 1. The difference between AQIs from different stations

Conventional dispersion models proposed in the environmental engineering are usually based on empirical assumptions and parameters that may not be applicable to different urban environments [3]. The crowd sensing-based method [1] could work for a very few kinds of gas like CO_2 but not applicable to aerosols, such as $PM_{2.5}$ and PM_{10}. The devices for detecting these pollutants are not easily portable and usually need a relatively long sensing period (e.g., 1~2hours) before generating an accurate AQI. Recently, big data reflecting city dynamics have become widely available [7], e.g., traffic flow, human mobility, and meteorology, enabling us to solve this challenging problem from a data perspective.

In this paper, we present a system that provides people with the real-time and fine-grained air quality throughout a city using a "cloud + client" architecture. In the system, the cloud infers the air quality of a location based on the (historical and real-time) air quality data from existing monitor stations and other relevant data sets, such as meteorology, traffic flow, structure of road networks, and POIs, we observed around the location. Using machine

learning and data mining techniques, we build a network between air quality labels and features observed across these data sources. The system also provides a mobile client, with which a user can monitor the air quality of multiple fine-grained locations in a city (e.g. the current location, home and work places) on a smart phone, and a web service that allows other applications to call the air quality of any location. The system can inform people's decision making, e.g., where and when to go jogging, and help diagnose the root cause of air pollution.

This paper introduces the implementation of the system, as the inference model of this system has been evaluated in paper [6]. The contribution of this paper lies in the following three aspects:

- We propose a hybrid framework (i.e., local servers + a cloud) to quickly enable a research prototype on the cloud in a cost-efficient way. This framework leverages the stability of a cloud platform to receive instant data, perform inferences, and provide online services, while using local servers for training models and maintaining data sources that do not change frequently. This framework saves storages and CPU resources on the cloud tremendously (i.e., lowers the monetary cost), while providing a certain flexibility to a research prototype's development (e.g., testing different parameters for a model is much more convenient in local servers than in the cloud). Our system has been deployed on Azure, a cloud service operated by Microsoft, providing the real-time and fine-grained air quality of nine Chinese cities, including Beijing, Shanghai, Shenzhen, and Guangzhou, etc.
- We devise a mobile client and a website that allow a user to monitor the air quality of any location. The mobile client and website communicate with the cloud via a web service. The mobile client (entitled Urban Air) can be installed via Window Phone App store. The website is hosted on Azure, publicly accessible via http://urbanair.msra.cn/

2. FRAMEWORK

As shown in Figure 2, our system consists of three major parts: local servers, the cloud, and consumers (e.g. mobile clients and websites), resulting in online and offline data flows, respectively. The local servers store static data sets, such as POIs, and train the inference model periodically, e.g., every month. The Cloud receive instant data, including meteorological and traffic data, infers air quality of each location every hour, and serve consumers with the inferred results via a web service. The consumers access the air quality data, displaying them on mobile clients or websites.

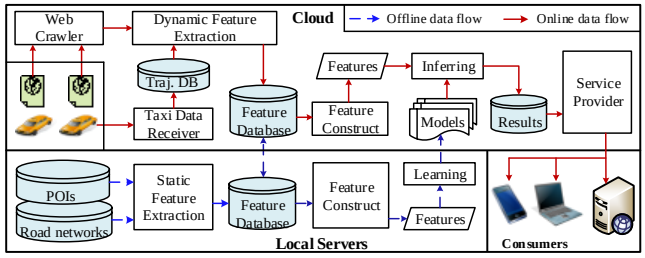

Figure 2. Framework of our system

2.1 The Cloud

The cloud crawls air quality readings of existing monitoring stations and meteorological data (such as weather conditions, humidity, and barometer pressure) from public websites every hour. The cloud also continuously receives GPS trajectories of taxicabs and then maps each trajectory onto a road network using a map-matching algorithm [5]. To save the resource on the cloud (more storages call for more expensive payment), we only store the recent trajectories in an online trajectory database. Historical trajectories can be moved to local servers periodically. The cloud extracts meteorological features from the data crawled from the web, and human mobility and traffic features from taxi trajectories. The taxi trajectory used here is just optional and better to have. Without the data, the inference model can still achieve an accuracy over 0.75.

The extracted features are stored in the online feature database. As there are different kinds of features, e.g., POI features and meteorological features, we need to conduct some feature combination for a location before feeding them into the inference model. Note that we do not simply put together different features into a single feature vector and treat them equally. Instead, they will be fed into different parts of our model, and combined in different ways (refer to Section 3 for details). Given the features of a location, the cloud infers its air quality and then stores the results in a database, which will be later accessed by mobile clients or websites via a web service.

We use Azure platform as a service (Paas). Table 1 details the Azure resources for our system. The web crawler and inference model share a small virtual machine (who has 1 core and 1.75GB memory), as they only work for a while in an hour. The website and web service share a medium virtual machine (A2), given the potential heavy accesses by many consumers. As the hybrid framework stores static data (like POIs) and historical trajectory data in local servers, 5GB is enough for storing the online features and inferred results of 9 cities. The expense for the total cloud resources is about 350USD per month.

Table 1. The Azure resources used for our system

Components	Azure Solution	Resources
Web Crawler & Inferring	Worker Role	Small (A1) 1 Core, 1.75 GB
Website & Web service	Web Role	Medium (A2) 2 Cores, 3.5 GB
Databases	SQL Azure	5GB

2.2 Local Servers

Basically, all the jobs can be done in the cloud if we do not consider the expenses. However, using cloud services, we need to pay for CPU hours, storages, and I/O bandwidths. Saving unnecessary cost is vital for a research prototype. Additionally, migrating big data from local servers up to the cloud is time-consuming given the limited network bandwidth. For instance, the size of the POIs and road networks data can be hundreds of Giga bytes, leading to a very long period of time (e.g., a few weeks) for copying the data from local servers to the cloud.

Given the above mentioned reasons, we propose a hybrid framework that combines local servers with the cloud. Specifically, we can extract features from POI and road network datasets offline and then inset the features into the online feature database. As the size of features is much smaller than that of raw data, a lot of storage and transferring time can be saved. In addition, the value of the two datasets does not change over time frequently. Thus, we can update the corresponding features every season. Likewise, we can train the inference model offline and update the online model periodically, e.g., every month. As the dynamic features are extracted in the cloud, we sync up the online feature database to local servers before each training process. In this way, we are agile to try new ideas (e.g. re-train the model) while significantly reducing expense for a research prototype.

2.3 Consumers

Figure 3 depicts the user interfaces of the mobile client. As demonstrated in Figure 3 A), a user has selected four locations, such as home and work places, to monitor on her mobile phone.

Here, each banner represents one location and the number shown in each banner is the AQI of the location. The color of a banner is determined in accordance with its air quality, e.g., "green" means a "good" and "yellow" denotes "moderate" in Chinese AQI standard. Each location was selected by long pressing the corresponding venue on a map, as shown in Figure 3B), where an icon stands for a venue that a user has selected. Our mobile client will automatically name a selected venue according to the titles of POIs and road networks around the venue. Users can then modify the name to some semantic title, such as home. By clicking a banner in the location list, users can also see the trend of air quality of a location, as illustrated in Figure 3 C). Once the air quality of these location exceeds a certain threshold, a user will be alerted by her mobile phone, hence then call her parents to close the windows or turn on air filters. The fine-grained air quality information can also inform a user's decision making on where and when to go jogging.

A) Location list **B) Select a location** **C) Trend of air quality**

Figure 3. Mobile User Interface

Figure 4 presents the user interface of the website, where an icon stands for an air quality monitoring station that has been built by governments and the number associated with an icon denotes the AQI of the station. Likewise, the color of an icon is set in accordance with the AQI of the station (refer to the colored bar descriptor shown on the bottom right part of Figure 4). The top right box of Figure 4 shows the average AQI well as the humidity and wind speed of a city. The box also presents the accuracy of the inferred AQIs in the city in past 48 hours. To validate the accuracy of our inference, we deliberately remove one station from the labeled data and predict the air quality of the station with our method. The reading from the station is then used as a ground truth to measure the inference results. We do such evaluate for each station in each hour, finally calculating an average accuracy over a period of time. The website covers 9 cities in China (the figure after the name of each city is the number of the monitoring stations in the city). We can switch between cities by clicking cities' name shown in the list.

The toolbar floating on the top-left part of Figure 4 helps us interact with the map. The most left button turns on and off the traffic flow that is overlapped on the map (this is to help diagnose the correlation between traffic and air quality through the exploratory visualization). The next three buttons respectively offer us a capability to see the air quality of a point location, in a spatial range, and throughout a city. For instance, as shown in the bottom-left part of Figure 4, we can view the air quality of any location (marked as a blue balloon icon) by just clicking on the map, even if there is no monitoring station. Once clicking on the third button, a user will see the results shown in Figure 5. The fourth button displays the top 200 locations in a city with the best and worst air quality over a period (e.g., in the past year). The information can inform a user's decision making, e.g., when purchasing a real estate. Clicking the last button, we will see the statistics on the air quality of the recent 60 days, as illustrated in Figure 6, where the top-left and top-right diagrams visualize the proportion of different AQI classes in daytime and nights, respectively. Generally, the air quality in the night is better than that of daytime in Beijing, as we can observe more green areas in the top-right chart. The bottom three figures presents the average AQI of an entire day of three pollutants respectively.

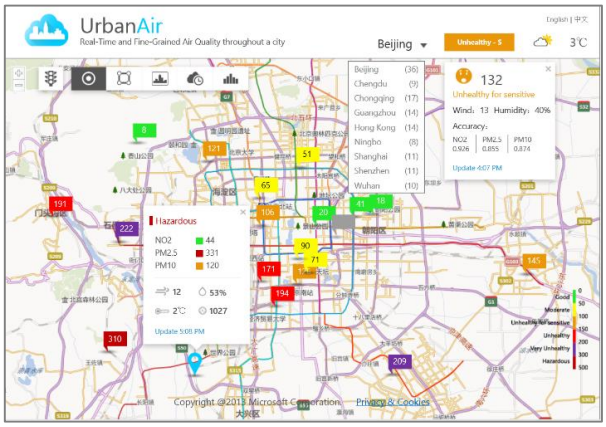

Figure 4. Web user interface

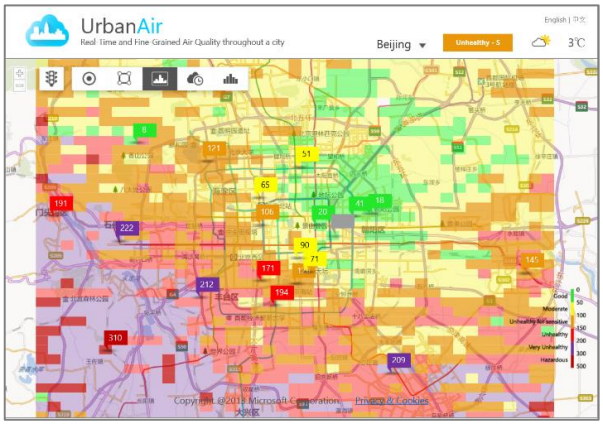

Figure 5. Fine-grained air quality throughout a city

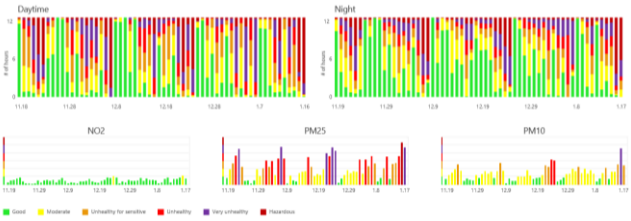

Figure 6. Statistics on the air quality data in recent 60 days

The mobile clients and website communicate with the cloud via a web service, following the data flow shown in Figure 7. To reduce the response time to a request, we load the inferred air quality of the recent hour from the results database to Azure virtual machine's memory. Two sets of APIs are defined for mobile clients and websites, respectively based on SOAP and HTTP protocols. An internal interface is designed to receive and answer requests through the APIs, retrieving the results according to the requests from the memory.

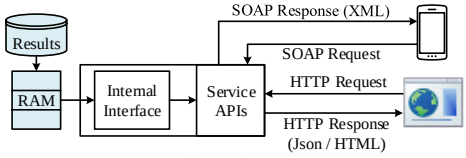

Figure 7. Data flow of the service provider

3. Learning and Inference

The inference model was proposed in our previous publication [4]. Here, we just give a very brief introduction to make this demo paper self-contained.

We divide a city into disjointed grids (e.g., 1km×1km), assuming the air quality in a grid cell is uniform while that of different grid cells may be different. If having an air quality monitor station, a grid cell is labeled by the AQIs reported from the station. We extract five categories (i.e., traffic, meteorological, human mobility, POI, and road network) of features respectively from the corresponding data observed in the cell and its eight surrounding cells. The output of the model is a class of air quality, consisting of good, moderate, unhealthy for sensitive group, unhealthy, very unhealthy, and hazardous (we use Chinese AQI standard, e.g., 0-50 means good). We train the model with labeled data and infer the grid cells without a monitoring station. After inferring the AQI category of a location, we further interpolate the real AQI value of the location based on the readings of the top three monitoring stations that are the geospatially closest to the location and have the same category of the AQI class as the location.

As we only have a few air quality stations in a city while there are many places to infer, the data with labels are very few. To address this issue, we propose a co-training-based semi-supervised learning approach, where unlabeled data are used to improve the inference accuracy. As shown in figure 8 A), a circle denotes a location and a plane means the states of these locations at a timestamp. We build two classifiers, a spatial classifier (*SC*) and a temporal classifier (*TC*), separately modeling the temporal dependency of air quality in an individual location and the spatial correlation of air quality among different locations. The two classifiers have a mutually reinforced learning in the framework of co-training [1].

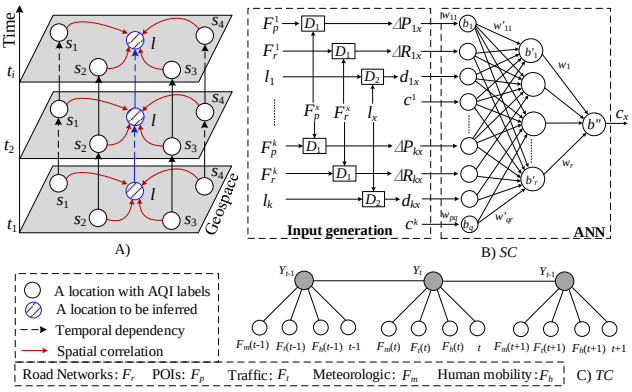

Figure 8. The philosophy of the inference model

The spatial classifier uses static features (e.g. POIs) to model the non-linear spatial correlations among air qualities of different locations. As illustrated in Figure 8 B), the *SC* consists of two parts: input generation (in the left box) and an artificial neural network, where F_p^k, F_r^k, l^k, and c^k denotes the POI features, road network features, location, and the AQI label of grid k; x is the grid to be inferred; D_1 is a distance function between features (e.g., the Pearson correlation in the experiments) and D_2 calculates the geo-distance between the center of two grids. We randomly choose n grid cells with labels to pair with the cell to be inferred (e.g. n=3 achieves the best accuracy in the experiments). To learn the impact of different scales of the distance between grids, we perform this pairwise process m times to formulate a collection of inputs. In the inference process, we also pair a grid to be inferred with a certain sets of n labeled grids, generating a prediction of AQI label for each set. The frequency of each inferred label is then used as the probability score of the label, and the most frequent label will be selected as the prediction result of SC.

The temporal classifier is based on a linear-chain conditional random field (CRF), which uses dynamic features (such as meteorology) to estimate the temporal transformation of air quality in a location. Figure 8 C) shows the graphical structure of the temporal classifier, which consists of hidden state variables **Y** and observations **X** (t is a timestamp by hour, e.g., 8am). At the inference time, we apply SC and TC to the corresponding features separately, selecting the most possible AQI class by the product of the two probability scores generated by the two classifiers. As different air pollutants (e.g., NO_2 and PM_{10}) are influenced by these factors differently, we build a model for each pollutant.

Figure 9 shows the performance of our method (U-Air) which outperforms six baselines, consisting of linear and Gaussian interpolations, classical air pollutant dispersion models, Decision Tree, CRF, and ANN. We add an instance into the training data if *SC* or *TC* predicts it as a class with a probability score over 0.85. As a result, as shown in the right part of Figure 9, the unlabeled data gradually improves the inference performance. Using a small Azure virtual machine, we can infer the air quality of entire Beijing in 3 minutes.

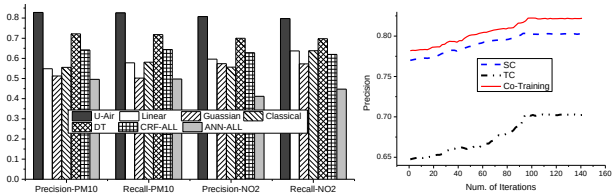

Figure 9. Overall results of different methods for PM_{10}

4. CONCLUSION

This paper presents a cloud-based system that provide users with real-time and fine-grained air quality throughout a city. The systems consists of a cloud, local servers, and consumers (including a mobile client and website). The hybrid framework that combines a cloud platform and local servers significantly saves monetary costs for a research project and bring flexibility for quickly trying new ideas, therefore can be referenced by other research projects if aiming to use a cloud platform as a service. Our cloud service is running on Windows Azure; the mobile client is available in Window Phone App store; the website is public accessible via http://urbanair.msra.cn/. The finer-grain air quality can inform people's decision making when jogging and cycling, and is also a step towards diagnose the root cause of air pollution.

5. REFERENCES

[1] Y. Jiang, K. Li, L. Tian, R. Piedrahita, X. Yun, O. Mansata, Q. Lv, R. P. Dick, M. Hannigan, and L. Shang. Maqs: A personalized mobile sensing system for indoor air quality. In Proc. of UbiComp 2011.

[2] K. Nigam, R. Ghani. Analyzing the Effectiveness and Applicability of Co-Training. In Proc. of CIKM 2000.

[3] S. Vardoulakis, B. E. A. Fisher, K. Pericleous, N. Gonzalez-Flesca. Modelling air quality in street canyons: a review. Atmospheric Environment 37 (2003) 155-182.

[4] J. Yuan, Y. Zheng, X. Xie. Discovering regions of different functions in a city using human mobility and POIs. In Proc. of KDD 2012.

[5] J. Yuan, Y. Zheng, C. Zheng, X. Xie, G. Sun. An Interactive-Voting based Map Matching Algorithm. In Proc. of MDM 2010.

[6] Y. Zheng, F. Liu, H.P. Hsieh. U-Air: When Urban Air Quality Inference Meets Big Data. In Proceeding of SIGKDD 2013.

[7] Y. Zheng, Y. Liu, J. Yuan, X. Xie. Urban Computing with Taxicabs. In Proc. of UbiComp 2011.