Spatially Fine-grained Urban Air Quality Estimation Using Ensemble Semi-supervised Learning and Pruning

Ling Chen[1], Yaya Cai[1], Yifang Ding[1], Mingqi Lv[2], Cuili Yuan[1], and Gencai Chen[1]
[1]College of Computer Science, Zhejiang University, Hangzhou, China
[2]College of Computer Science, Zhejiang University of Technology, Hangzhou, China
[1]{lingchen, caiyaya, dyf_716, clyuan, chengc}@zju.edu.cn
[2]lvmingqi1104@163.com

ABSTRACT

Air pollution has adverse effects on humans and ecosystem, and spatially fine-grained air quality information (i.e., the air quality information of every fine-grained area) can help people to avoid unhealthy outdoor activities. However, the number of air quality monitoring stations is usually limited, and thus spatially fine-grained air quality estimation is a challenging task. This paper proposes a method for inferring spatially fine-grained air quality information throughout a city. On one hand, since air quality is affected by multiple factors (e.g., factory waste gases and automobile exhaust fumes), this method employs various data sources, including traffic, road network, point of interests (POIs), and check-ins from social network services, which are related to air quality, to conduct the estimation. On the other hand, since the labeled data are highly limited due to the sparseness of monitoring stations, this method uses an improved ensemble semi-supervised learning (Semi-EP) to establish the relationship between the various data sources and urban air quality. Semi-EP firstly generates multiple classifiers from the original labeled data set and these classifiers are retrained in the iterative co-training process. Then, ensemble pruning technique is used to select the most-diverse subset from these multiple classifiers. This method is evaluated on the real-world dataset of Hangzhou city, China, and the experimental results have demonstrated its advantages over state-of-the-art methods.

Author Keywords

air quality estimation; urban computing; data mining; semi-supervised learning; ensemble learning.

ACM Classification Keywords

H.5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

UbiComp '16, September 12-16, 2016, Heidelberg, Germany
© 2016 ACM. ISBN 978-1-4503-4461-6/16/09…$15.00
DOI: http://dx.doi.org/10.1145/2971648.2971725

INTRODUCTION

Recently, some developing countries (e.g., China and Brazil) are suffering from air pollution, which has adverse effects on humans and ecosystem. The spatially fine-grained air quality information (i.e., the air quality information of every fine-grained area) is useful to the public, e.g., people can exploit the information to arrange their outdoor activities. However, obtaining such information is a challenging task, and the reasons are as follows. On one hand, air pollutant concentrations (e.g., $PM_{2.5}$ and PM_{10}) vary spatially significantly with anthropogenic emissions and meteorological conditions, because they are affected by multiple sources, e.g., stationary sources (factories, power plants, and dry cleaners), dynamic sources (cars and buses), and naturally occurring sources (windblown dust and volcanic eruptions) [6, 8]. On the other hand, the number of air quality monitoring stations is usually limited by the extensive cost, size, and bulk of different sensors. For example, China has 1436 stations in 338 prefecture-level cities by the end of 2014, which means that each city has only 4 stations on average.

Existing works try to address this problem using different models, e.g., physical models, linear statistical models, and non-linear statistical models. Physical models simulate the actual physical dispersions of air pollutants, which are usually based on a number of empirical assumptions that are not easy to conduct [9, 26]. For example, Gaussian Plume model assumes that the concentrations of air pollutant are dispersed in the vertical and horizontal directions in a Gaussian manner and wind speed equals to 1m/s at the stack tip (50m) [2]. Some Street Canyon models [16, 24] require the emission density (per street length and per time), street width, and vertical dispersion parameters of the receptor point.

Linear statistical models, e.g., Land-Use Regression (LUR) and Kriging, are also widely used to estimate spatially fine-grained air quality [20]. LUR model measures the local land-use features (e.g., traffic count, road type, terrain elevation, and distance to pollutant source) of air quality monitoring stations [5, 14, 18, 25] and then employs linear regression models to identify the relation between the air quality and the local land-use features. Shad *et al.* [27] estimated the concentration of PM_{10} at unmonitored places using fuzzy genetic linear membership Kriging, which is a spatial interpolation method. Linear statistical models are used to discover linear relationships rather than non-linear relationships. Since air quality depends on multiple factors, it varies by locations non-linearly, and thus the performance of linear approaches is not satisfactory.

There are also several non-linear air quality estimation models. For example, Hasenfratz *et al*. [12] used Generalized Additive Models (GAMs), which can find non-linear relationships between air quality and explanatory variables, to estimate pollution concentrations at unmonitored places based on land-use and traffic data. Jutzeler *et al*. [15] developed a region-based model that employs Gaussian process regression (GPR) framework to estimate urban air pollution dispersion in Zurich metropolitan area.

All of the above mentioned statistical models have a common problem, i.e., they need labeled data to build data-driven models. Since the number of available monitoring stations are usually limited in urban space, the lack of variety of labeled data would significantly degrade the performance of supervised methods. Aiming at this problem, Zheng *et al*. [33] proposed a co-training based method named U-Air to infer spatially fine-grained urban air quality. Co-training algorithm [3] is a classical ensemble semi-supervised learning method consisting of two classifiers, which improve the performance by using unlabeled data to augment labeled data.

However, only when data have two sufficient and redundant views (i.e., attribute sets), which are conditionally independent to each other, can co-training algorithm improve the classification performance. Aiming at this problem, several new ensemble semi-supervised learning methods are proposed, e.g., SemiBoost [21] and Multi-Semi AdaBoost (MSAB) [29], which use both classifier predictions and pairwise similarities to maximize the margin. Both of them do not require two sufficient and redundant views. By exploiting both manifold and cluster assumptions, SemiBoost aims at minimizing the empirical error on labeled data and the inconsistency on labeled and unlabeled data. SemiBoost is a binary classifier and MSAB is a multiclass classifier, which is an extension of SemiBoost by employing coding scheme.

The aforementioned ensemble semi-supervised learning methods all use a fixed threshold to select the high-confident pseudo-labeled examples for retraining. For example, U-Air selects all high-confident pseudo-labeled examples, whose labeling confidence is larger than 85%. SemiBoost selects the top 10% of the pseudo-labeled examples, and MSAB selects the top 15%. However, these simple pseudo-labeled example selection schemes have not considered the problem that whether the number of additional unlabeled examples is sufficient to compensate for the increase of classification noise. Thus, it would cause a problem that classification noise becomes larger with the increase of pseudo-labeled examples.

On the other hand, in order to capture the influence factors on air quality, most traditional methods consider geographic variables, e.g., road type, traffic count, elevation, and land cover. For example, distance to coast, distance to industrial source, and intense use land, etc., are considered in [20]. Demirbas *et al*. [5] considered street density and weather characteristics, etc., in the model. With the development of smart cities, various urban data related to air quality become available and could be utilized to estimate spatially fine-grained urban air quality. Zheng *et al.* [33] considered urban data that reflect city dynamics, e.g., POIs. In addition, the spatial correlations of air qualities between two areas is utilized to estimate the air quality. However, U-Air uses random selection scheme to search nearby areas having monitoring stations when modelling the spatial correlation of air qualities between two areas, which would cause inconsistency problem, because the characteristics order could influence the model.

Since human mobility implies traffic flow and land-use, which have impacts on air quality, it is related to the air quality. With the development of smartphones and social network services (SNSs), a large amount of people participate in social networks and many users like to tag specific locations by using social applications on their smartphones. These tags are called check-ins, which represent human mobility and might have potential correlation with air quality. Check-in data provide an opportunity to improve the performance of spatially fine-grained urban air quality estimation.

In this paper, an ensemble semi-supervised learning and pruning (called Semi-EP) based method is proposed to estimate spatially fine-grained urban air quality. We estimate the air quality throughout a city using air quality data reported by a limited number of monitoring stations, as well as data from various sources, including not only traffic, road network, and POIs used in U-Air, but also check-ins. The spatial correlation between an area and nearby areas having monitoring station is also utilized to estimate the air quality of the area. We use k nearest neighbour search rather than random selection to search nearby areas based on their geographical distance, road network, and POIs, which can eliminate randomness and solve inconsistent problem. Semi-EP firstly generates multiple classifiers from the original labeled data set. Then, these classifiers are retrained iteratively by exploiting unlabeled data, which are generated from unmonitored places in the city. Finally, the method selects the most-diverse subset of these multiple classifiers. In contrast to previous ensemble semi-supervised learning algorithms that use fixed threshold to decide the amount of pseudo-labeled examples added to the training data set at every iteration, Semi-EP considers the relation between the amount of pseudo-labeled examples and classification noise, which guarantees that the amount of additional unlabeled examples is sufficient to compensate for the increase of classification noise. In summary, the contributions of the paper are listed as follows:

1) Propose a spatially fine-grained urban air quality estimation method based on semi-supervised learning and various urban data. On one hand, semi-supervised learning could utilize unlabeled data to improve the estimation accuracy, so its performance would better than supervised learning when limited number of air quality monitoring stations are available. On the other hand, the various urban data (including traffic, road network, POIs, and check-ins) could help to capture the influence factors on air quality.

2) Present an improved ensemble semi-supervised learning algorithm, which employs more classifiers to gain better generalization ability and to make pesudo-labeled examples more confident. In addition, ensemble pruning technique is used to select the most-diverse subset from multiple classifiers.

3) Exploit the spatial correlation of the air qualities of two areas to improve air quality estimation accuracy, which uses k nearest neighbour search instead of random selection to search nearby air quality monitoring stations to solve inconsistent problem.

The rest of the paper is organized as follows. Section 2 describes a spatially fine-grained urban air quality estimation method using ensemble semi-supervised learning with ensemble pruning. Section 3 presents performance evaluations, and Section 4 finally concludes the paper.

METHODOLOGY

Partition

The fine-grained area is represented by grid cell, which is the basic unit to perform urban air quality estimation, and we suppose that the air quality in a grid cell is uniform. A city is partitioned into many disjointed grid cells, and we estimate air quality for each one of them separately. We extract local features for each grid cell g from its affecting area denoted by $g.A$, which consists of g and g's eight neighbour grid cells. As shown in Figure 1, the shadow area is a grid cell, and the larger square area with black border is its affecting area.

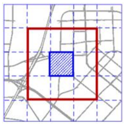

Figure 1. Grid cell and its affecting area.

Framework

The framework of our spatially fine-grained urban air quality estimation method is shown in Figure 2, which works based on three data flows: preprocessing, learning, and estimation.

In the preprocessing data flow, we firstly extract various features (including POI-related features F_p, road-network-related features F_r, traffic-related features F_t, and check-in features F_c) for each grid cell from urban data (i.e., POIs, road-network, traffic, and check-ins) observed in its affecting area. Note that F_p and F_r are spatially-related features, while F_t and F_c are temporally-related features. The detailed description of feature extraction is given in Section 2.3. Then, nearby monitoring-related features F_g are constructed based on the spatially-related features (F_p and F_r) and air quality labels. The detailed description of F_g construction is given in Section 2.3.5. Since air pollutant concentrations reported by air quality monitoring stations are numeric, they are transformed into nominal values that can be regarded as air quality labels. Finally, we get the feature vectors of each grid cell by putting F_g, F_t, and F_c together.

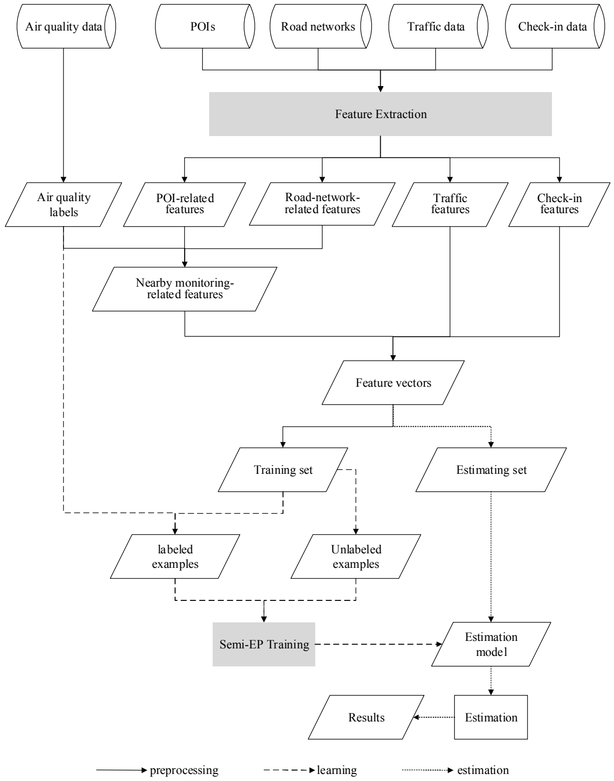

Figure 2. The framework of air quality estimation.

In the learning data flow, the feature vectors and air quality labels of grid cells having air quality monitoring stations are combined as labeled examples, while the feature vectors of grid cells that do not have air quality monitoring stations are used as unlabeled examples. Because there are usually limited air quality monitoring stations in a city, labeled examples are few while unlabeled examples are abundant. To counter the problem, the proposed semi-supervised method (the detailed description is given in Section 2.4) uses unlabeled examples to improve the performance of air quality estimation. The labeled examples and the unlabeled examples are taken as the input of Semi-EP. After iterative training, the estimation model is generated. Since air pollution factors have different impacts on different air pollutants, different estimation models are constructed for different air pollutants.

In the estimation data flow, the current feature vector of a grid cell is extracted and fed into the estimation models, and the current IAQIs (Individual Air Quality Index) of different air pollutants of the grid cell are estimated. IAQI is a number representing the level of each air pollutant, which indicates how polluted the air is. Note that the air quality data reported by air monitoring stations are updated in a fixed time interval. Therefore, the air quality estimation should be conducted periodically.

Feature extraction

In this section, we describe the extraction of five kinds of features (i.e., traffic-related features, road-network-related features, POI-related features, check-in features, and nearby monitoring-related features) from urban data.

Traffic-related features

Traffic flow is a major source of air pollutants in urban areas. The traffic data include the average vehicle speeds of all primary roads in a city, which are sampled at a fixed interval. Thus, we can obtain a vehicle speed time series T= {$r.v_1$, $r.v_2$…$r.v_t$… $r.v_n$} for each primary road r, where $r.v_t$ is the vehicle speed on road r at time t. We extract the following features for each grid cell g.

1) *Expectation of* vehicle *speed in the past n_e hours*: $E(V)$

First, we compute the average vehicle speed of each road in the past n_e hours by Eq. 1, where L denotes the length of the time series of one hour. Then, we calculate the expectation of the vehicle speed in the past n_e hours in $g.A$ by Eq. 2.

$$E(r_i.v) = \frac{\sum_{t=1}^{n_e \times L} r_i.v_t}{n_e \times L} \tag{1}$$

$$E(V) = \frac{\sum_{r_i \in g.A} E(r_i.v)}{\left|\{r \mid r \in g.A\}\right|} \tag{2}$$

2) Variance *of vehicle speed in the past n_d hours*: $D(V)$

We calculate the variance of vehicle speed by Eq. 3 to capture how far the vehicle speed is spread out in the past n_d hours in $g.A$.

$$D(V) = \frac{\sum_{r_i \in g.R} \sum_{t=1}^{n_d \times L} (r_i.v_t - E(V))^2}{\left|\{r \mid r \in g.A\}\right| \times (n_d \times L) - 1} \tag{3}$$

Road-network-related features

Road density is a surrogate for traffic flow that could contribute to local concentrations of air pollutants, and different road types might have different traffic patterns. Thus, we extract road-network-related features F_r by Eq. 4 for each grid cell, where $len(r \cap g.A)$ denotes the intersection length of r and $g.A$. $rtype$={h, s, t, r, f} is a set of road types, where h, s, t, r, f denote primary road, secondary road, tertiary road, residential road, and footway road, respectively. $g.R_h$={r | $r \in g.A$ ∧ $r.t$=h} is the set of primary roads in $g.A$, where $r.t$ denotes road type, i.e., $r.t \in rtype$. Likewise, $g.R_s$, $g.R_t$, $g.R_r$, and $g.R_f$ are secondary road set, tertiary road set, residential road set, and footway road set in $g.A$, respectively.

$$g.F_r = \left\{ \sum_{r \in g.A} len(r \cap g.A) \right\} \cup \left\{ \sum_{r \in g.R_i} len(r \cap g.A) \,\middle|\, i \in rtype \right\} \tag{4}$$

POI-related features

POI type and density can affect air quality indirectly. For instance, more chemical factories in an area can lead to worse air quality of the area. Since there are a large amount of POI types, we classify each POI as one of the categories shown in Table 1. For each grid cell g, we extract the amount of POIs belong to each category in $g.A$ by Eq. 5, where p and $p.ty$ denote a POI and its type, and $count$() is a counting function.

$$g.F_p = \left\{ \left\{ count(p) \,\middle|\, p.ty \in C_i \wedge p \in g.A \right\} \,\middle|\, i = 1, 2 \cdots, 10 \right\} \tag{5}$$

C_1: Transportation spots	C_6: Stadiums
C_2: Factories	C_7: Schools
C_3: Parks	C_8: Real estate
C_4: Stores	C_9: Entertainment
C_5: Eating and drinking establishment	C_{10}: Other establishment

Table 1. Category of POIs.

Check-in features

Since human mobility implies traffic flow and land-use, human mobility in an area is also strongly related to the air quality of the area. The amount of check-ins in social networking services is a surrogate for human mobility. Therefore, for each grid cell g, we extract the number of check-ins in the past n_c hours in $g.A$ by Eq. 6, where c denotes a check-in, and $c.t$ denotes the time interval between the time of c and the air quality estimation moment.

$$g.F_c = \left\{ count(c) \,\middle|\, c \in g.A \wedge c.t \le n_c \right\} \tag{6}$$

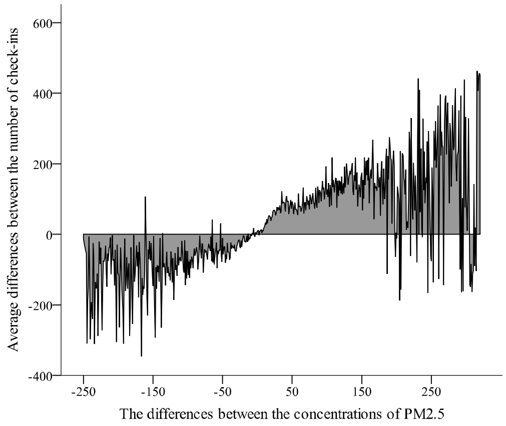

Figure 3. The correlation between two differences (the difference between the $PM_{2.5}$ concentrations of two grid cells, and the difference between the numbers of check-ins in these cells).

Figure 3 shows the correlation between the concentrations of $PM_{2.5}$ and the numbers of check-ins, where n_c is set to 12. Given two grid cells g_1 and g_2, it can be found that the concentration of $PM_{2.5}$ in g_1 is larger than that in g_2 (positive value in x axis) when the amount of check-ins in $g_1.A$ is larger than that in $g_2.A$ (positive value in y axis). Likewise, when the concentration of $PM_{2.5}$ in g_1 is smaller than that in g_2, the amount of check-ins in $g_1.A$ is smaller than that in $g_2.A$. This suggests that there has strongly correlation between the differences of the concentration of $PM_{2.5}$ and the amount of check-ins in the past 12 hours.

Nearby monitoring-related features

Generally, the air quality of an area is strongly related to its nearby areas. Therefore, given a grid cell, the air quality of its nearby monitor stations and their correlations, which are calculated over their geospatial features, are extracted.

Given two grid cells g_t and g_p, U-Air [33] calculated the geographical distance between their centres (denoted by d_{tp}), Pearson correlation between their road-network-related features (denoted by R_{tp}), and Pearson correlation between their POI-related features (denoted by P_{tp}). If g_t is to be estimated, d_{tp}, R_{tp}, P_{tp}, and the air quality of g_p are extracted and added to the feature vector of g_t. Because the air quality of g_p should be known, g_p is selected from grid cells having monitoring station, where k grid cells are randomly selected as g_p.

Although this pairwise process is performed multiple times in the training phase to learn the impact of distance scale, it would cause inconsistency problem. For example, suppose that k is set to 3, grid cell groups (g_1, g_2, g_3) and (g_2, g_3, g_4) are randomly selected for g_t at the same time. Because the same value in different positions of a feature vector plays different roles in the learning algorithm, g_2 and g_3 would play different even opposite roles in different feature vectors during training phase, and thus such a design would reduce the performance of the model.

Therefore, we employ k nearest neighbour search approach instead of randomized policy to select k grid cells. The k grid cells are sorted in ascending order by their distance from g_t. The distance between two grid cells g_t and g_p is measured by Euclidean Distance over d_{tp}, R_{tp},P_{tp},T_{tp}, and C_{tp}, where T_{tp} and C_{tp} are the differences of traffic-related features and check-in features, respectively.

Semi-EP Training

In this section, we describe the proposed Semi-EP training method, which is composed of an ensemble semi-supervised learning and an ensemble pruning algorithm.

Ensemble Semi-supervised Learning

The proposed ensemble semi-supervised learning is originated from co-training [3], which requires two sufficient and redundant views of the data. The idea of co-training algorithm is as follows: Given a labeled data set represented as L, it firstly learns two different models for the two views. Then, the algorithm improves the classification accuracy iteratively. At each iteration, each classifier labels unlabeled examples and adds high-confident newly labeled examples to L. After the update of L, the two classifiers are trained again.

In most real-world applications, the data set might not have two different views that are redundant but completely not correlated. Therefore, the standard co-training algorithm usually does not perform well. To counter the problem, learning multiple classifiers with a single view has been developed, e.g., COM [10] and Tri-training [35].

Our ensemble semi-supervised learning algorithm also employs multiple learning algorithms for a single view and refines each classifier at each iteration. We apply n classifiers $E = \{C_1, C_2, \ldots, C_n\}$ in the co-training style model, where C_i maps the feature space to the label space. Figure 4 shows the procedure of our algorithm at the tth iteration. L_i denotes the labeled data set for C_i and L_i is generated through bootstrap sampling from L; $F_{i,t}$ is the collection of high-confident unlabeled examples that are labeled for C_i at the tth iteration; U is the collection of unlabeled examples; $U_{i,t}$ is the subset of $F_{i,t}$ after the selection process, which is put into training set L_i at the next iteration; $U_{i,0}$ is set to Ø, i.e., there are no unlabeled examples labeled for C_i at the first iteration.

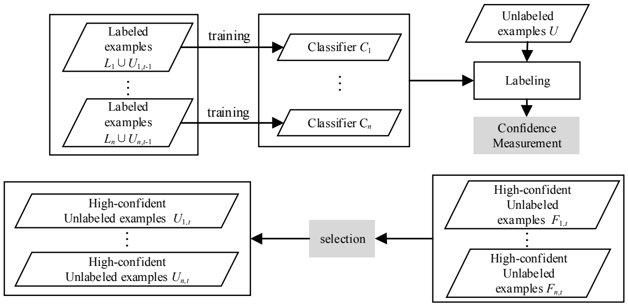
Labeled examples $L_1 \cup U_{1,t-1}$
Labeled examples $L_n \cup U_{n,t-1}$
training
training
Classifier C_1
Classifier C_n
Unlabeled examples U
Labeling
Confidence Measurement
High-confident Unlabeled examples $U_{1,t}$
High-confident Unlabeled examples $U_{n,t}$
selection
High-confident Unlabeled examples $F_{1,t}$
High-confident Unlabeled examples $F_{n,t}$

Figure 4. Procedure of ensemble semi-supervised algorithm at *t*th iteration.

At the tth iteration, each classifier C_i is trained on dataset $L_{i,t} \cup U_{i,t\text{-}1}$. Then, the pseudo-label and its confidence of each unlabeled example are computed and the high-confident unlabeled examples are put into $F_{i,t}$. Next, $U_{i,t}$ is generated after the selection of $F_{i,t}$ by some criteria. Confidence measurement and selection, which are in the grey shaded areas of Figure 4, are two most important phases in our algorithm. These two fundamental phases will be described next.

In the confidence measurement phase, the confidences of unlabeled examples need to be measured to generate the set of high-confident unlabeled examples F. Ineffective confidence estimation would lead to performance issue, because many incorrect labeled examples would be involved into the retraining phase.

In our method, the pseudo-label of an unlabeled example x_u is assigned to classifier C_i by majority voting from all other classifiers, denoted as $E_i = \{C_j \in E | j \neq i\}$. The confidence of

x_u can be estimated by the degree of agreements on the labeling from E_i according to Eq.7, where $h_i(x_u)$ is the classification result of example x_u based on C_i, and $\hat{l}^i_{x_u}$ is the pseudo-label assigned to x_u by E_i via majority voting strategy. I is an indicator function, i.e., if x is true, $I(x) = 1$, otherwise $I(x) = 0$.

$$conf_i(x_u) = \frac{\sum_{\mathrm{j}=0,\, j\neq i}^{n} I\left(h_j(x_u) = \hat{l}^i_{x_u}\right)}{n-1} \tag{7}$$

By using this approach, if the majority of classifiers in E_i make a correct prediction on x_u, C_i would receive a helpful rather than noisy new example for retraining.

Note that, Tri-training receives x_u for C_i as long as E_i reach a consensus on labeling x, regardless of the opinion of C_i. In order to effectively improve the performance of C_i, we use a "majority teaches minority" strategy. To be specific, only those newly-labeled unlabeled examples, on which C_i disagrees with E_i, are considered. x_u is regarded as a high-confident example if its labeling confidence is larger than a pre-set threshold θ. Then, $F_{i,t}$, a set of high-confident examples for C_i at the tth iteration, can be obtained by Eq. 8.

$$F_{i,t} = \left\{x \mid conf_i(x) \geq \theta \ \ \Lambda \ \ h_i(x) \neq \hat{l}^i_x, x \in U\right\} \tag{8}$$

When a classifier has not captured the underlying pattern of the data distribution, especially in the initial iteration, misclassification by E_i is most likely to happen. It would import noisy data and result in performance degradation. Thus, the algorithm should guarantee that the amount of additional unlabeled examples is sufficient to compensate for the increase of the classification noise. Angluin and Laird [1] found the relationship, i.e., Eq. 9 (equivalently Eq. 10), between the hypothesis worst-case classification error rate e, classification noise rate η and sample size m, where other parameters (e.g., the number of hypothesis) are held constant c. This relationship can be used to control classification noise rate.

$$m = \frac{c}{e^2(1-2\eta)^2} \tag{9}$$

$$e = \sqrt{\frac{c}{m(1-2\eta)^2}} \tag{10}$$

Eq. 10 implies that the performance of the algorithm can be improved by having a larger value of $m(1\text{-}2\eta)^2$. Through the formula reduction [35], Eq. 11 and Eq.12 should be satisfied in order to reduce the classification error rate of C_i at the tth iteration, where $m_{i,t}$ is the size of $U_{i,t}$ and $e_{i,t}$ is the classification error rate over $U_{i,t}$. Because the real labels of the unlabeled examples are unknown, we compute $e_{i,t}$ in the initial labeled data set L by Eq. 13, where $|\cdot|$ means the size of the collection and l_x is the real label of x. In other word, $e_{i,t}$ is the prediction error rate of E_i on the data set L.

$$0 < \frac{e_{i,t}}{e_{i,t-1}} < \frac{m_{i,t-1}}{m_{i,t}} < 1 \tag{11}$$

$$m_{i,t} = \left\lceil \frac{e_{i,t-1} m_{i,t-1}}{e_{i,t}} - 1 \right\rceil \tag{12}$$

$$e_{i,t} = \frac{\left|\left\{x \middle| \ conf_i(x) \geq \theta \ \ \Lambda \ \ h_i(x) \neq \hat{l}^i_x \ \ \Lambda \ \ \hat{l}^i_x \neq l_x\right\}\right|}{\left|\left\{x \middle| conf_i(x) \geq \theta \ \ \Lambda \ \ h_i(x) \neq \hat{l}^i_x\right\}\right|} \ , \ x \in L \tag{13}$$

Therefore, the maximum size of $U_{i,t}$ is determined by Eq. 12, and Eq. 11 is used as the stopping criterion for our method, i.e., Semi-EP jumps out of the iterative process when Eq. 14, Eq. 15, or Eq.16 is not satisfied. Because $e_{i,t}$ is at least 0 and $m_{i,t}$ is at most $|U|$, the iteration procedure would converge eventually. Note that $U_{i,t\text{-}1}$ added to the training set at the (t-1)*th* iteration would be removed from the training set and put back to U before starting the tth iteration.

$$e_{i,t} < e_{i,t-1} \tag{14}$$

$$m_{i,t} > m_{i,t-1} \tag{15}$$

$$e_{i,t} m_{i,t} < e_{i,t-1} m_{i,t-1} \tag{16}$$

Because the size of $F_{i,t}$ is usually larger than $m_{i,t}$, we should remove ($|F_{i,t}|$-$m_{i,t}$) examples from $F_{i,t}$ to satisfy Eq.12. Tri-training employs a random strategy, which does not take into account how much C_i disagrees on the labeling of the unlabeled examples in $F_{i,t}$. Obviously, how much C_i disagrees on the unlabeled examples x_u is in direct proportion to the probability score of classifying x_u into class label $h_i(x_u)$ by C_i, which is denoted by $P(h_i(x_u))$, and how much E_i agrees on x_u is in direct proportion to $conf_i(x_u)$. In our algorithm, the high-confident examples x_u of $F_{i,t}$ are firstly sorted by $conf_i(x_u)$ in descending order, when coincides, sorted by $p(h_i(x_u))$ in descending order. After ranking, the top $m_{i,t}$ newly labeled examples are put into $U_{i,t}$ for further training.

Ensemble Pruning

Diversity among the classifiers in an ensemble learning method has been recognized as a key characteristic to the success of classifier combination [17, 30, 34]. There are two strategies applied in our method to enhance diversity. The first one is that different supervised learning algorithms are employed. The other one is that the collection of high-confident unlabeled examples for C_i is completely disjoint from other classifiers' by Eq.7. In addition, $L_{i,t}$ is generated through bootstrap sampling from L. Thus, at the tth iteration, C_i is trained on data set $L_{i,t} \cup U_{i,t}$ to introduce randomness into the input of the learning algorithm.

However, at each iteration, the ensemble semi-supervised learning makes classifiers more similar by the "majority teaches minority" strategy, which might reduce the diversity among the classifiers. In addition, the large number of ensemble members could lead to extra computational cost and memory usage. It might not always be true that the larger the size of ensemble is, the better the performance it has [32]. Thus, ensemble pruning technique is employed in our method to search a good subset of ensemble members that maximizes the diversity.

An ensemble pruning framework (PEP) proposed by [23] achieves these two objectives (maximizing the diversity and minimizing the number of members in ensemble) via an evolutionary Pareto optimization method. The framework is incorporated into our method.

Given a set of n trained component classifiers $E = \{C_1, C_2, \ldots, C_n\}$ after applying the ensemble semi-supervised learning, and let H_s denotes a pruned ensemble with the selector vector $s \in \{0,1\}^n$, where s_i equals to 1 means that the ith component classifier C_i is selected, the bi-objective ensemble pruning problem is formulated as Eq. 17, where f denotes the performance measure function, e.g., error rate on training set [19] and the diversity measure of H_s. Many measures of the relationship between two classifier outputs can be derived from the statistical literature [17], which are regarded as the diversity of two classifiers, e.g., Q statistics [31], disagreement measure [13], and double-fault measure [7]. Detailed algorithms about PEP can be found in [23].

$$\arg\min_{s\in\{0,1\}^n} \left(f\left(H_s\right), |s|\right) \tag{17}$$

After ensemble pruning, final hypothesis is generated by majority voting, which is widely used in ensemble learning.

EXPERIMENTS

Dataset

The dataset used in our experiments consists of traffic, road network, POIs, Weibo check-ins, and air quality data from Hangzhou city, China. Weibo is a micro blogging software, by which some people enjoy tagging their current locations. These tags are called Weibo check-in data. All air monitoring stations of Hangzhou have the same sensors and report the concentrations of all pollutants. There are six stations in Hangzhou city, the data of these six stations are regarded as labeled data, while the data of other grid cells without monitoring stations are regarded as unlabeled data. Traffic, Weibo check-in, and air quality data have been collected from November 1, 2013 to September 1, 2014. Due to data missing problem, the effective time of a grid cell is 4500 hours on average. There are 27421 labeled examples and 2886170 unlabeled examples. Detailed information about the data set is as follows:

1) *Traffic Data*: The traffic data is collected from the website of Hangzhou Transportation Research Center[1]. The update time interval is 5 minutes.

2) *Road Networks*: The road network data is downloaded from OpenStreetMaps[2] (OSM).

3) *POIs*: Due to the sparseness of POIs on OSM, the POIs of Hangzhou city are collected by Google Place API.

4) *Weibo check-in Data*: Weibo API is used to obtain the number of check-ins every one hour in each $g.A$.

5) *Air Quality Data*: The concentrations of various air pollutants, e.g., $PM_{2.5}$, PM_{10}, and NO_2 etc., are collected from a public website[3] every hour.

IAQI ranging from 1 to 6 represents the level of each air pollutant. The larger value of IAQI indicates worse air quality. We transform the concentration of air pollutants into IAQI labels, i.e., class labels, by the standard issued by the Ministry of Environmental Protection of the People's Republic of China [22]. The distributions of six class values are as follows: 17.6%, 39.5%, 21.8%, 8.8%, 9.0%, and 3.3%.

Experimental setup

Since we assume that the concentrations of air pollutants in a grid cell are uniform, there is an issue in determining the grid cell size. Zheng *et al.* [33] divided a city into disjointed rectangular grid cells, each one with size 1 km × 1 km, which is employed in our experiments.

In the ensemble pruning phase of our method, diversity measure is used as performance measure function. The more diverse the ensemble is, the smaller the error is. We measure diversity based on Q-statistics [31], which is one of the most popular diversity measures and is formalized based on pairwise difference between every pair of individual classifiers. Given two classifiers C_i and C_k, their Q-statistics and diversity are measured by Eq. 18 and Eq. 19, respectively, where N_{11} is the number of examples that C_i and C_k recognize both correctly, i.e., $N_{11}=|\{x|h_i(x) = h_k(x) = l_x\}|$; N_{10} is the number of examples that C_i recognizes correctly while C_k recognizes incorrectly, i.e., $N_{10}=|\{x|h_i(x) =l_x \wedge h_k(x)\neq l_x\}|$; N_{00} and N_{10} are calculated in the same way. Note that, we exploit not only labeled data, but also unlabeled data to measure $D_{i,k}$. Given an ensemble E of n classifiers, the diversity of E is calculated by Eq. 20.

$$Q_{i,k} = \frac{N_{11}N_{00} - N_{01}N_{10}}{N_{11}N_{00} + N_{01}N_{10}} \tag{18}$$

$$D_{i,k} = 1 - Q_{i,k} = \frac{2N_{01}N_{10}}{N_{11}N_{00} + N_{01}N_{10}} \tag{19}$$

$$D_E = \frac{2}{n \times (n-1)} \sum_{i=1}^{n-1} \sum_{k=i+1}^{n} D_{i,k} \tag{20}$$

Because the number of unlabeled examples with the highest confidence value (i.e., 1) is always larger than the limited number of noisy examples in the experiments, the confidence threshold θ in Eq. 8 has no effect in the selection phase of our method. Through parameter tuning, n_e, n_d, n_c about feature extraction are set to 10, 10, and 12, respectively.

[1] http://www.hzjtydzs.com/web/current.aspx

[2] http://www.openstreetmaps.org/

[3] http://www.pm25.com/city/hangzhou.html

To evaluate the performance of the estimation methods, we take the data of a monitoring station as test data, and the data of other monitoring stations as training data. Since there are six monitoring stations in total, the procedure is repeated six times to evaluate the model and the average classification accuracy is reported.

Results and discussion

Evaluation on features

To evaluate the effectiveness of different features, we remove some features and evaluate the corresponding accuracies via J48 decision tree (J48) that is significantly sensitive to features. In the experiments, we use the WEKA implement of J48 with default parameter settings [11].

The results are shown in Table 2. It can be seen that J48 with all features performs the best. Note that when F_c (check-in features) is added to feature set F_t+F_r+F_p, the classification accuracies on $PM_{2.5}$, PM_{10}, and NO_2 are respectively raised to 0.624, 0.628, and 0.636 from 0.609, 0.611, and 0.594, which suggests that check-in information can improve the estimation performance. In addition, we can find that the result on NO_2 is better than that on $PM_{2.5}$ and PM_{10} in general. The reason for this is probably that $PM_{2.5}$ and PM_{10} are affected by many complex factors, e.g., dust, pollen, and tephra, which are not identified in the model and these factors have little influence on the concentration of NO_2.The data set of $PM_{2.5}$ is used in the following experiments.

Features	$PM_{2.5}$	PM_{10}	NO_2
F_t	0.443	0.446	0.494
F_c	0.413	0.422	0.469
F_r	0.416	0.403	0.438
F_p	0.437	0.430	0.426
F_r+F_p	0.522	0.518	0.513
F_r+F_t	0.513	0.526	0.528
F_r+F_c	0.539	0.523	0.538
F_c+F_p	0.548	0.540	0.535
F_c+F_t	0.545	0.552	0.560
F_p+F_t	0.586	0.582	0.577
F_t+F_r+F_p	**0.609**	**0.611**	**0.594**
F_c+F_r+F_p	0.547	0.556	0.562
F_c+F_r+F_t	0.576	0.578	0.582
F_c+F_p+F_t	0.595	0.601	0.603
F_t+F_r+F_p+F_c	**0.624**	**0.628**	**0.636**

Table 2. Results related to features for $PM_{2.5}$, PM_{10}, and NO_2.

K nearest neighbour search VS random selection

To evaluate the effectiveness of the *k* nearest neighbour search approach that is used to select *k* nearby grid cells described in Section 2.3.5, we compare it with the random selection approach. The value of *k* varies from 1 to 5. The size of the ensemble is also taken into account to verify the effectiveness of the *k* nearest neighbour search approach under different ensemble sizes. Semi-EP-*n* denotes Semi-EP method whose ensemble size is *n*. J48 is used as the base classifier in this experiment.

Figure 5 shows that the optimum value of *k* is 3 for the *k* nearest neighbour search approach, while the optimum value of *k* is 4 for the randomized policy. In addition, the highest classification accuracy of *k* nearest neighbour search approach (*k* = 3) is higher than that of random selection (*k* = 4), whatever the size of the ensemble is, which suggests that the performance of *k* nearest neighbour search approach is better than random selection.

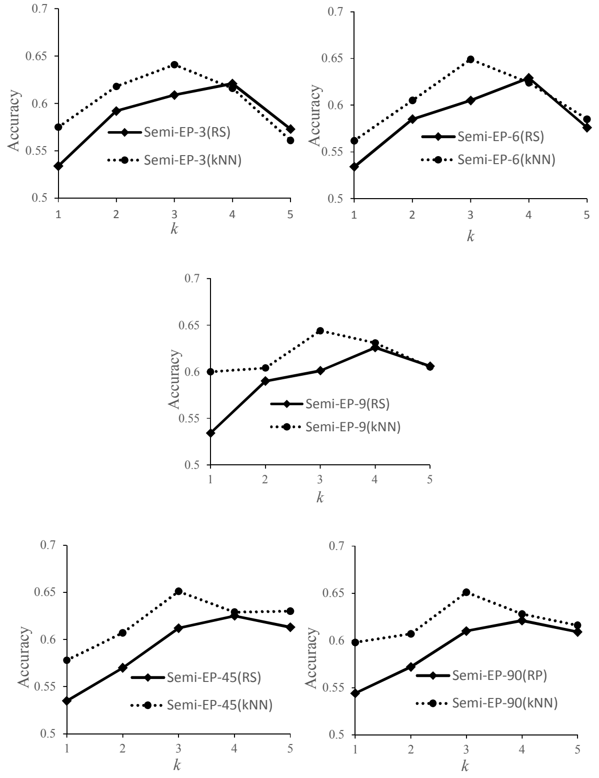

Figure 5. Accuracy comparison for the random selection (RS) and the *k* nearest neighbor search (*k*NN) under different k values.

Choice of Base Classifier

Figure 6 compares the supervised methods, i.e., J48, Naïve Bayes (NB), Random Forest (RF), and Conditional Random Field (CRF), with the Semi-EP method. CRF [28] is a discriminative undirected probabilistic graphical model that can model overlapping and non-independent features, and RF [4] is an ensemble method that introduces randomness in the tree learning process. The symbol EP-X denotes Semi-EP with three identical base classifiers X and symbol EP-X-Y-Z denotes Semi-EP with three different base classifiers X, Y, and Z.

The results indicate that Semi-EP improves the performance of all the four base classifiers and EP-CRF-RF-NB performs the best. In addition, we can see that when Semi-EP employs different base classifiers, i.e., EP-X-Y-Z, the performances are generally better than EP-X. It might be because different base classifiers introduce diversity that could decrease variance and avoid overfitting. We chose

CRF, RF, and NB as the base classifiers in the following experiments, and the size of the ensemble of our method is a multiple of 3.

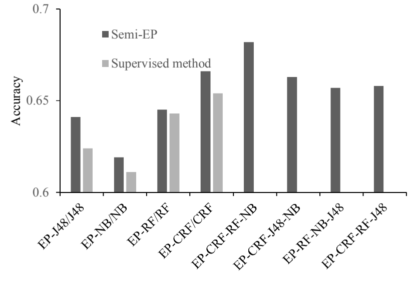

Figure 6. The classification accuracies of Semi-EP and its base classifiers.

The impact of ensemble size

To study the impact of the ensemble size on the performance, the classification accuracies under different ensemble sizes are compared. The results are shown in Figure 7. It can be seen that when n equals to 12, Semi-EP performs the best. The classification accuracy tends to become stable with the increase of n when n> 24. Therefore, we set the size of the ensemble to 12, i.e., n =12.

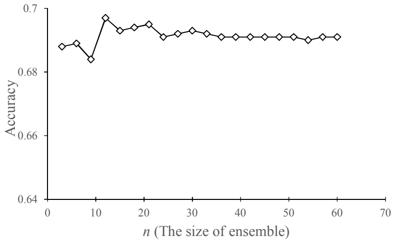

Figure 7. Accuracy over different ensemble size.

The impact of the number of selected high-confident unlabeled samples

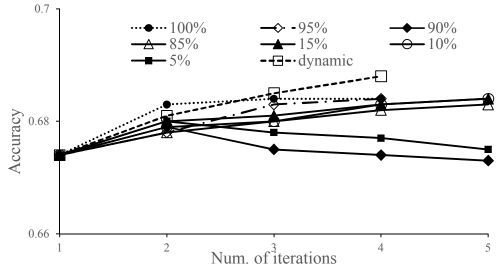

Figure 8. Learning progress of Semi-EP.

Figure 8 shows the performance of Semi-EP by varying the value of α, i.e., the proportion of high-confident pseudo-labeled examples selected for retraining during the iterations. 'dynamic' denotes the value of α is derived from Eq. 12 at each iteration and is not a fixed number. It can be seen that the performances of Semi-EP are improved at the second iteration and decrease at subsequent iterations when α is set as 90% and 5%. It might be because the number of additional unlabeled examples is insufficient to compensate for the increase of the classification noise. When α is set to 10%, 15%, 85%, 95%, 100%, and 'dynamic' respectively, the classification accuracies are improved as the increase of iteration. However, the amount of improvement varies by these settings and 'dynamic' performs the best. It indicates that our strategy is effective.

Comparison with other semi-supervised methods

We compare the proposed ensemble semi-supervised learning algorithm (Semi-EP) with the four aforementioned ensemble semi-supervised methods (SemiBoost, MSAB, Co-training, and Tri-training) under different base classifiers (J48, RF, NB, and CRF). In addition, to statistically measure the significance of performance difference, pairwise t-tests at the 5% level are carried out between Semi-EP and other methods. Note that Co-training uses two feature sets, i.e., temporal features ([F_t, F_c]) and spatial features ([F_g]), while other ensemble semi-supervised learning algorithms use single feature set ([F_t, F_c, F_g]), where F_g denotes nearby monitoring-related features. Since SemiBoost is a binary classifier, one-vs-all method is employed to handle the multiclass classification problem. Based on the results of pervious experiments, n in Semi-EP is set to 12. We also tune the parameters of other algorithms to output their best performance.

Table 3 shows the classification accuracies of different ensemble semi-supervised learning algorithms using different base classifiers. Since Semi-Boost, MSAB, and Co-training should have the same base classifiers, their accuracies in the last column are set to 'empty'. It can be seen that the overall classification accuracies of Semi-EP and Tri-training are higher than that of other algorithms. Note that Semi-EP and Tri-training perform the best when CRF-RF-NB is used as their base classifiers. The advantage of Semi-EP and Tri-training over other algorithms might be because they limit the amount of newly labeled examples at each iteration as discussed in Section 2.4.1, which takes the performance of classifier into account.

The results also show that when J48 is used as base classifier, Tri-training outperforms Semi-EP. However, Semi-EP outperforms Tri-training when RF, NB, CRF, and CRF-RF-NB are used as base classifiers. The advantages of Semi-EP over Tri-training can be concluded as follows. First, more classifiers are employed in the ensemble, which makes the high-confident unlabeled data more reliable. In addition, ensemble pruning technique is applied to select the most-diverse subset of the ensemble, which can eliminate the negative effects of large ensemble size. Second, when measuring the confidence of pseudo-labeled examples for a classifier, whether a classifier disagrees with other classifiers is taken into account by Semi-EP. Third, when selecting high-confident pseudo-labeled examples, Semi-EP firstly sorts the examples according to their labeling confidence and the degree of misclassification. Then it selects the top pseudo-labeled examples instead of randomly selection that Tri-training uses.

Algorithms	J48	RF	NB	CRF	CRF-RF-NB
SemiBoost	0.615±0.012*	0.604±0.009*	0.625±0.005*	0.613±0.006*	-
MSAB	0.616±0.063*	0.601±0.016*	0.609±0.011*	0.609±0.009*	-
Co-training	0.608±0.009*	0.629±0.008*	0.613±0.006*	0.591±0.002*	-
Tri-training	**0.647±0.002**	0.635±0.010*	0.626±0.006*	0.655±0.001	0.679±0.004*
Semi-EP	0.641±0.011	**0.647±0.002**	**0.638±0.003**	**0.656±0.001**	**0.688±0.005**

Table 3. Accuracy comparison between different algorithms under different base classifiers (mean±std.). * indicates Semi-EP method is statistically superior to the compared method (pairwise t-test at the 5% level).

Comparison with other air quality estimation methods

Table 4 shows the classification accuracies of different air quality estimation methods. GPR is a non-parametric method that has been successfully applied in the last decades to various fields, e.g., geo-statistics and spatial statistics. We use continuous values in GPR and discretize the numeric result into IAQI labels for evaluation. U-Air is a state-of-the-art urban air quality estimation method using co-training-based semi-supervised learning, which consists of a temporal classifier (CRF) and a spatial classifier (Back Propagation Neural Network, BPNN). CRF is employed to model the temporal dependency of air quality based on temporal features ([F_t, F_c]) and BPNN is employed to model the spatial correlation of air quality based on spatial features ([F_g]). Note that original U-Air method randomly selects the nearby monitoring stations to construct F_g. Thus, its F_g is different from that of other air quality methods. U-Air(-F_c) denotes a variance of U-Air method that does not utilize check-in features. U-Air(+*k*NN) denotes a modified U-Air method using *k* nearest neighbour search while constructing F_g.

Methods	Accuracy
GPR	0.607±0.012*
U-Air	0.633±0.007*
U-Air(-F_c)	0.617±0.009*
U-Air(+*k*NN)	0.642±0.003*
Our Method	**0.688±0.005**

Table 4. Accuracy comparison between different air quality estimation methods (mean±std.). * indicates our method is statistically superior to the compared method (pairwise t-test at the 5% level).

Table 4 shows that the classification accuracy of our method is higher than GPR and U-Air, and the performance of U-Air is better than that of U-Air(-F_c), which proves that check-in features play a role in air quality estimation. In addition, the classification accuracy of U-Air(+*k*NN) is higher than that of U-Air, which suggests that using *k* nearest neighbour search instead of random selection while constructing F_g works better.

The advantage of our method over GPR is that our method exploits unlabeled data and ensemble technique to improve model performance, especially when there are few stations in a city (lack of diversity and abundance of labeled data). The advantage of our method over U-Air can be concluded as follows. First, U-Air requires data have two sufficient and redundant views. The requirement might not be satisfied in the experiment and thus the performance would be affected. Second, in the iterative process of co-training, U-Air suffers from noisy pseudo-labeled examples, while our method utilizes noise theory to guarantee that the amount of pseudo-labeled examples is sufficient to compensate for the increase of the classification noise. Third, while constructing F_g, U-Air uses random selection rather than *k* nearest neighbour search to select *k* nearby stations. The performance of these two approaches is compared in Section 3.3.2, which demonstrates the advantage of *k* nearest neighbour searching approach.

CONCLUSIONS AND FUTURE WORK

In this paper, we proposed an ensemble semi-supervised learning and pruning based method to estimate spatially fine-grained urban air quality based on air quality data from a few monitoring stations and air quality-related features (i.e., traffic-related features, road-network-related features, POI-related features, and check-in features) extracted from various urban data. We firstly evaluated the effectiveness of these features using data obtained in Hangzhou for $PM_{2.5}$ and PM_{10}. The experimental results demonstrate the effectiveness of these features. The other experiments, e.g., comparing the performance between spatially fine-grained urban air quality estimation methods, are also conducted on the data set of Hangzhou. The experiments show the following results. First, using *k* nearest neighbour search approach to find nearby grid cells having station is better than random selection approach. Second, check-in information can improve the estimation performance. Third, the fixed proportion of pseudo-labeled examples put into labeled data set would suffer from noisy examples.

In the future, we would like to apply our method to more cities to make the experiments more convincing. In addition, we will try to take account of the size of grid cells, which might affect the performance of spatially fine-grained air quality estimation. Furthermore, we will try to employ our method with forecasting features to forecast air pollution.

ACKNOWLEDGMENTS

This work was funded by the Ministry of Industry and Information Technology of China (No. 2010ZX01042-002-003-001), China Knowledge Centre for Engineering Sciences and Technology (No. CKCEST-2014-1-5), the National Natural Science Foundation of China (Nos. 60703040 and 61332017), the Science and Technology Department of Zhejiang Province (Nos. 2011C13042 and 2015C33002), Zhejiang provincial natural science foundation of China (No. LY15F020025).

REFERENCES

1. Angluin, D., Laird, P. Learning from noisy examples. *Machine Learning*, 2, 4 (1988), 343-370.
2. Arystanbekova, N. K. Application of Gaussian plume models for air pollution simulation at instantaneous emissions. *Mathematics and Computers in Simulation*, 67, 4 (2994), 451-458.
3. Blum, A., Mitchell, T. Combining labeled and unlabeled data with co-training. In *Proc. of the 11th Annual Conference on Computational Learning Theory* (1998), 92-100.
4. Breiman, L. Random forests. *Machine Learning*, 45, 1 (2001), 5-32.
5. Demirbas, M., Rudra, C., Rudra, A., *et al*. Imap: Indirect measurement of air pollution with cellphones. In *Proc. of the 7th Annual IEEE International Conference on Pervasive Computing and Communications* (2009), 1-6.
6. Elliot, P., Wakefield, J. C., Best, N. G., *et al*. Spatial epidemiology: methods and applications. *Oxford University Press* (2000).
7. Giacinto, G., Roli, F. Design of effective neural network ensembles for image classification processes. *Image Vision and Computing Journal*, 19, 9 (2001), 699-707.
8. Gilliland, F., Avol, E., Kinney, P., *et al*. McConnell. Air pollution exposure assessment for epidemiologic studies of pregnant women and children: lessons learned from the Centers for Children's Environmental Health and Disease Prevention Research. *Environmental Health Perspectives* (2005), 1447-1454.
9. Godish, T., Fu, J. S. Air quality. *CRC Press* (2003).
10. Goldman, S., Zhou, Y. Enhancing supervised learning with unlabeled data. In *Proc. of the 17th International Conference on Machine Learning* (2000), 327-334.
11. Hall, M., Frank, E., Holmes, G., *et al*. The WEKA data mining software: an update. *ACM SIGKDD Explorations Newsletter*, 11, 1 (2009), 10-18.
12. Hasenfratz, D., Saukh, O., Walser, C., *et al*. Pushing the spatio-temporal resolution limit of urban air pollution maps. In *Proc. of the 12th Annual IEEE International Conference on Pervasive Computing and Communications* (2014), 69-77.
13. Ho, T. The random space method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20, 8 (1998), 832-844.
14. Hoek, G., Beelen, R., De Hoogh, K., *et al*. A review of land-use regression models to assess spatial variation of outdoor air pollution. *Atmospheric Environment*, 42 (2008), 7561-7578.
15. Jutzeler, A., Li, J. J., Faltings, B. A Region-Based Model for Estimating Urban Air Pollution. In *Proc. of the 28th AAAI Conference on Artificial Intelligence* (2014), 424-430.
16. Kim, M. J., Park, R. J., Kim, J. J. Urban air quality modeling with full O3–NOx–VOC chemistry: Implications for O3 and PM air quality in a street canyon. *Atmospheric Environment*, 47 (2012), 330-340.
17. Kuncheva, L. I., Whitaker, C. J. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine Learning*, 51, 2 (2003), 181-207.
18. Larson, T., Henderson, S. B., Brauer, M. Mobile monitoring of particle light absorption coefficient in an urban area as a basis for land use regression. *Environmental Science & Technology*, 43, 13 (2009), 4672-4678.
19. Margineantu, D. D., Dietterich, T. G. Pruning adaptive boosting. In *Proc. of the 14th International Conference on Machine Learning* (1997), 211-218.
20. Mercer, L. D., Szpiro, A. A., Sheppard, L., *et al*. Comparing universal krigingand land-use regression for predicting concentration of gaseous oxides of nitrogen (NOx) for the Multi-Ethnic Study of Atherosclerosis and Air Pollution(MESA Air). *Atmospheric Environment*, 45 (2011), 4412-4420.
21. Mallapragada, P., Jin, R., Jain, A., *et al*. Semiboost: Boosting for semi-supervised learning. *Pattern Analysisand Machine Intelligence*, 31, 11 (2014), 2000-2014.
22. Ministry of Environmental Protection of the People's Republic of China (MEP). Technical Regulation on Ambient Air Quality Index. *China Environmental Science Press* (2012).
23. Qian, C., Yu, Y., Zhou, Z. H. Pareto Ensemble Pruning. In *Proc. of the 29th AAAI Conference on Artificial Intelligence* (2015), 2935-2941.
24. Rakowska, A., Wong, K. C., Townsend, T., *et al*. Impact of traffic volume and composition on the air quality and pedestrian exposure in urban street canyon. *Atmospheric Environment*, 98 (2014), 260-270.
25. Ross, Z., English, P. B., Scalf, R., *et al*. Nitrogen dioxide prediction in Southern California using land use regression modeling: potential for environmental health analyses. *Journal of Exposure Science and Environmental Epidemiology*, 16, 2 (2006), 106-114.
26. Scaar, H., Teodorov, T., Ziegler, T., *et al*. Computational Fluid Dynamics (CFD) Analysis of Air Flow Uniformity in a Fixed-Bed Dryer for Medicinal Plants. In *Proc. of the 1st International Symposium on CFD Applications in Agriculture*, 1008, 4 (2013), 119-126.

27. Shad, R., Mesgari, M. S., Shad, A. Predicting air pollution using fuzzy genetic linear membership kriging in GIS. *Computers, Environment and Urban Systems*, 33, 6 (2009), 472-481.

28. Sutton, C., Mccallum, A. An Introduction to Conditional Random Fields. *Foundations & Trends® in Machine Learning*, 4, 4 (2010), 93-127.

29. Tanha, J., Someren, M. V., Afsarmanesh, H. Boosting for Multiclass Semi-Supervised Learning. *Pattern Recognition Letters*, 37, 1 (2013), 63-77.

30. Wu, X., Kumar, V., Quinlan, J. R., *et al*. Top 10 algorithms in data mining. *Knowledge and Information Systems*, 14, 1 (2008), 1-37.

31. Yule, G. U. On the association of attributes in statistics: with illustrations from the material of the childhood society. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 194 (1900), 257-319.

32. Zhang, Y., Burer, S., Street W N. Ensemble pruning via semi-definite programming. *The Journal of Machine Learning Research*, 7 (2006), 1315-1338.

33. Zheng, Y., Liu, F., Hsieh, H. P. U-Air: when urban air quality inference meets big data. In *Proc. of the 19th International Conference on Knowledge Discovery and Data Mining* (2013), 1436-1444.

34. Zhou, Z. H. When semi-supervised learning meets ensemble learning. *Frontiers of Electrical and Electronic Engineering in China*, 6, 1 (2011), 6-16.

35. Zhou, Z. H., Li, M. Tri-training: Exploiting unlabeled data using three classifiers. *IEEE Transactions on Knowledge and Data Engineering*, 17, 11 (2005), 1529-1541.