A Weighted Crowdsourcing Approach for Network Quality Measurement in Cellular Data Networks

Yaliang Li, Jing Gao, *Member, IEEE*, Patrick P. C. Lee, *Member, IEEE*, Lu Su, *Member, IEEE*, Caifeng He, Cheng He, Fan Yang, and Wei Fan

Abstract—With ubiquitous smartphone usages, it is important for network providers to provide high-quality service to every user in the network. To make more effective planning and scheduling, network providers need an accurate estimate of network quality for base stations and cells from the perspective of user experience. Traditional drive testing approach provides a quality measurement for each area and the quality measurement is obtained from the equipment in a moving vehicle. This approach suffers from the limitations of high costs, low coverage, and out-of-date values. In this paper, we propose a novel crowdsourcing approach for the task of network quality estimation, which incurs little costs and provides timely and accurate quality estimation. The proposed approach collects quality measurements from individual end users within a certain network or cell coverage area, and then aggregates these measurements to obtain a global measurement of network quality. We propose an effective aggregation scheme which infers the information weights of end users and incorporates such weights into the estimation of network quality. Experiments are conducted on two datasets collected from citywide 3G networks, which involve $616,796$ users and $22,715$ cells. We validate the effectiveness of the proposed approach compared with baseline method. From the aggregated measurement results, we observe some interesting patterns about network quality, which can be explained by network usage and traffic behavior. We also show that proposed approach runs in linear time.

Index Terms—Crowdsourcing, network quality measurement, cellular network

1 Introduction

Nowadays, smartphones are playing indispensable roles in people's lives. People use them to communicate and share items with friends through social networks, watch real-time videos, send and receive emails, etc. With the explosion of smartphone users, third generation (3G) cellular networks have been deployed widely over the world, and the transition to the fourth generation (4G) has started. The ultimate goal of network providers is to provide high-quality network experience to all the users. One key factor in providing high-quality network service is the ability of accurately measuring network quality in real-time so that actions can be taken immediately when network quality drops.

In this paper, we study the problem of measuring network quality for network providers. Specifically, we hope to provide the quality measurement of a base station or a cell. From the perspective of users, the network quality may vary within the range of a base station or a cell. However, network providers need a single indicator to represent the network quality of a base station or a cell. Once such indicator results are available, network providers could use such information to answer important questions in planning and scheduling: Which area should have more base stations? How to decide and optimize the choice of cells for users across the entire network to balance loads on different cells? How to plan in advance to avoid bad network quality and heavy loads based on historical data of network quality?

The traditional way used in practice to assess network quality is "drive testing" [1], in which technicians drive a vehicle carrying the equipment to measure various parameters and derive the quality of networks in a particular area. This approach may consume extensive resources and human labors. Since it takes time and efforts to drive around, this approach cannot cover a large area at the same time or adapt to the change of network quality in real-time.

To overcome these limitations, we propose a novel crowdsourcing approach to derive accurate and reliable estimates of network quality for a given area by *aggregating information from a crowd of users*. Specifically, network providers can easily collect network quality measurement from each user [2], [3]. For a given area, we can aggregate measurements obtained from all the users within this area and output a final estimate of the area network quality. Compared with the traditional drive testing approach, our proposed crowdsourcing approach has the following advantages: (1) The cost of collecting measurement data from users is much smaller than that incurred in drive

- *Y. Li, J. Gao, and L. Su are with the University at Buffalo, The State University of New York, 338 Davis Hall, Buffalo, NY 14260. E-mail: {yaliangl, jing, lusu}@buffalo.edu.*
- *P. P. C. Lee is with the Department of Computer Science and Engineering, The Chinese University of Hong Kong, Shatin, Hong Kong. E-mail: pclee@cse.cuhk.edu.hk.*
- *C. He and C. He are with the Huawei Noahs Ark Lab, 525-520, 5/F, Core Building 2, Hong Kong Science Park, Shatin, Hong Kong. E-mail: {hecaifeng, hecheng}@huawei.com.*
- *F. Yang is with the Tencent Holdings Limited, Shenzhen, Guangdong, China. E-mail: fineyang@tencent.com.*
- *W. Fan is with the Baidu Research Big Data Lab, 1195 Bordeaux Drive, Sunnyvale, CA 94089. E-mail: fanwei03@baidu.com.*

Manuscript received 21 Apr. 2015; revised 27 Feb. 2016; accepted 15 Mar. 2016. Date of publication 25 Mar. 2016; date of current version 5 Jan. 2017.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TMC.2016.2546900

testing. (2) As users span all geographical areas, the proposed approach can easily cover all the areas simultaneously in network quality estimation. (3) By controlling the frequency of information collection, we can provide real-time network quality estimates and monitor network quality changes timely. (4) Besides providing network quality estimates of an area, the proposed approach can also output user usage patterns that cannot be obtained by drive testing. Such patterns are invaluable for network providers for future planning and quality control.

The major technical challenge in the crowdsourcing approach is how to conduct effective aggregation among diverse quality measurements collected from a large amount of users. The naive way to aggregate the network quality information from users is to conduct voting, in which the measurement value that is claimed by the majority of users is selected as output. The drawback of this simple aggregation is that they treat all the user measurement information equally. This may not be effective in real practice as the collected information can be affected by various user-related factors. For example, the following user-related factors may lead to the degradation of network quality: bad hardware quality in some users' devices, frequent transitions between cells resulted by users' highly mobile moving patterns, some personal habits such as constantly refreshing websites or Apps that result in a high volume of web requests. All of these user-related factors affect their user experience and the information collected from them. Thus when aggregating measurement information from various users, it is important to *treat different users differently by considering the effect of these user-related factors*.

Therefore, instead of conducting voting without distinguishing users, we should adopt a weighted aggregation scheme which takes user-related factors into consideration when aggregating network quality measurements collected from different users. Here the user weight indicates how much information from this particular user should be taken into consideration for the aggregation after removing user-related factors. If a user has a big bias in his measurements (e.g., his measurements are heavily affected by user-related factors), the corresponding weight will be low, while a high weight indicates the corresponding user has a small bias in his measurements. Unfortunately, user-related factors are not easy to quantify, thus user weights are not known *a priori*.

To tackle this challenge, we propose an effective way to estimate user weights and aggregate network quality simultaneously based on the following principle: Information from a user with a high weight should be counted more in the aggregation to derive the aggregated quality of network, and a user whose provided information is close to the aggregated result should have a high weight. An optimization function is defined based on this principle to aggregate user measurements for better estimation of network quality. In this optimization framework, user weights and aggregated measurements are defined as two sets of unknown variables, and they are jointly estimated to optimize the objective function. By iteratively updating one set of variables based on the other set, the estimation on user weights and aggregated measurements are mutually enhanced until convergence is reached.

To evaluate the proposed crowdsourcing aggregation approach for assessing cellular network quality, we collect data from two citywide 3G networks, which totally involve $616,796$ users and $22,715$ cells. Data are collected during an eight or four days period from measurement report log files, IP packet traces, performance call history and record log files. From these files, we extract a collection of network quality measurements from individual users about the cells they are using. We experimentally examine the variety of user weights on these two datasets to verify the benefit of the proposed weighted aggregation scheme. The effectiveness of weighted aggregation via user weight estimation is demonstrated by comparing with majority voting baseline on simulations and case studies. Then we derive network quality using the proposed aggregation scheme on these two datasets, and illustrate various patterns in network quality based on the aggregation results. These patterns are further explained by correlating with network usage and traffic behavior. Moreover, we show that the proposed aggregation approach is efficient and linear in running time, and thus can be deployed in large-scale network for real-time scenario.

To sum up, in this paper, we propose to deliver an effective solution to the network quality measurement task. We provide an end-to-end solution to network providers by extracting measurements from users' data and aggregating their measurements. More specifically, the contributions of this paper are:

- We present a novel, effective and cost-efficient crowdsourcing approach to measure network quality as an alternative to the drive testing approach. By wisely utilizing crowd information, the proposed approach can provide timely and accurate estimate of network quality, which is important for many planning and scheduling tasks of network providers.
- Instead of applying simple voting, we propose to integrate user measurements by a weighted combination scheme in which weights capture user-related factors. We propose an optimization method to jointly estimate user weights and aggregated network quality. By taking into account user weights, the aggregated measurements are more precise.
- We conduct thorough experimental analysis on two real world datasets involving $616,796$ users and $22,715$ cells, which are collected recently from two citywide 3G operation networks. The results demonstrate the effectiveness of the proposed weighted aggregation scheme.
- We analyze the time complexity of the proposed method, and experimentally confirm that it runs in linear time with respect to the number of observations. We also implement the parallel version of the proposed method based on Hadoop, which makes the proposed method suitable for large-scale real-time quality monitoring.

2 METHODOLOGY

In this section, we first formally define the task of aggregating user measurement for network quality estimation. A majority voting baseline method is then described. After that, we present the proposed aggregation method, which conducts weighted voting via user weight estimation. We formulate the task as an optimization problem by considering the effect of user-related factors. Finally, we discuss some practical issues to adapt the proposed model to different task scenarios and platform settings.

2.1 Problem Formulation

To formally define the problem, we start by introducing some important terms:

- *object*: An object is the target whose network quality we want to measure. It could be a radio network controller, a base station, or a cell, each of which can cover a set of users. In this task, we focus on cells. As network quality and users in the cell are both changing over time, we are interested in monitoring the *status* of network quality for an object, which is defined as the network quality of *an object at a certain time*. For example, "a cell with ID 37601 at $19:28$pm, $21/07/2013$" denotes a status whose network quality is to be evaluated. We represent a status using variable o, which corresponds to an object at a certain time. We denote a set of statuses as $\mathcal{O} = \{o_1, o_2, \ldots, o_N\}$, which contains a set of objects monitored at different time stamps.
- *user*: A user is an end user who is supported by the network, which is denoted as u. Each status o has a group of users, denoted by $\mathcal{U}_o$, which is the set of users who are using the object at a particular time. We define the set of users for all statuses as $\mathcal{U} = \cup_{i=1}^{N} \mathcal{U}_{o_i}$.
- *claim*: A user u makes a claim about a status o when the user is supported by the object's network at a particular time. We collect the measurement information about this network's quality from this user. The measurement value is denoted as v_o^u. As the requirement from network provider, in the following we assume the network quality measurements are discrete, that is, v_o^u represents a value chosen from predefined set $\mathcal{S} = \{\text{"}good\text{"}, \text{"}medium\text{"}, \text{"}bad\text{"}\}$. We will discuss more details about this in Section 3.
- *value*: For a status o, each user u in the set $\mathcal{U}_o$ provides a measurement *value* v_o^u to indicate network quality, and we combine the information from all the users in $\mathcal{U}_o$ to get an *aggregated value* v_o^* as the final output. In other words, v_o^* is the desired measurement of the network quality at status o.
- *weight*: Each user $u \in \mathcal{U}$ has a weight t_u, which indicates how much of the information from this particular user should be taken into consideration for aggregation. A low user weight suggests that the information from this user is highly affected by user-related factors and we should not rely on the information from this user when we calculate the aggregated value, while the information from a user with a higher weight will be counted more in the aggregation as this user has small bias (e.g., user-related factors) in his measurement.

To sum up, in this task, suppose there are N statuses to be monitored, each of which has a corresponding set of users $\mathcal{U}_o$. From each user u in the set, a network quality measurement value v_o^u is collected for the status o. The objective is to integrate measurement values $\{v_o^u\}_{u \in \mathcal{U}_o}$ among users, and get an aggregated measurement v_o^* for each status o.

2.2 Baseline Method

A naive solution for this aggregation task is to find the value that is claimed by the majority of the users. That is, for each status o, the value $v \in \mathcal{S}$ that has the highest count among users is selected as the final output v_o^*:

$$v_o^* = \arg\max_{v \in \mathcal{S}} \sum_{u \in \mathcal{U}_o} \mathbb{1}(v_o^u = v). \tag{1}$$

Although this approach provides an efficient way to aggregate user measurement values, it assumes that information from different users are equally important. It doesn't consider the effect of user-related factors and doesn't distinguish the difference among various users. However, as discussed in Section 1 and justified in Section 5.1, the quality measurements are affected by user-related factors and users should be distinguished.

2.3 Weighted Aggregation

It is more reasonable to conduct weighted aggregation of network quality measurements by considering the effect of user-related factors. Suppose the weight of each user, denoted as t_u, is available. We can incorporate such weights into the aggregation using weighted combination. The aggregated measurement v_o^* for each status o is calculated as:

$$v_o^* = \arg\max_{v \in \mathcal{S}} \sum_{u \in \mathcal{U}_o} t_u \cdot \mathbb{1}(v_o^u = v). \tag{2}$$

In this weighted voting scheme, we trust more the information v_o^u that is claimed by the users who have higher weights. In Section 5.2, our experimental results show that the weighted voting scheme outputs more reasonable measurement values compared with simple majority voting.

As weighted aggregation is a preferred aggregation scheme, the key is to derive users' weights to be incorporated in the aggregation. However, user-related factors are typically unknown, and when the aggregated network quality is unknown neither, the user weights cannot be easily estimated. To tackle this challenge, we propose a novel approach to estimate both user weights and network quality jointly. The basic principle is inspired by our previous work [4] : Information from a user with a high weight should be counted more in the aggregation, and a user whose provided information is close to the aggregated result should have a high weight. Based on this principle, we design a method that tightly integrates the process of network quality aggregation and user weight estimation. Specifically, we formulate the task as the following optimization problem:

$$\begin{aligned} \min_{\{v_o^*\},\{t_u\}} \quad & \sum_{o \in \mathcal{O}} \sum_{u \in \mathcal{U}_o} t_u \cdot dis(v_o^u, v_o^*) \\ \text{s.t.} \quad & t_u \geq 0, \quad h(\{t_u\}) = 1, \end{aligned} \tag{3}$$

where $dis(\cdot)$ is a distance function that measures the difference between two measurement values, and 0-1 loss function is chosen here: If v_o^u is the same as v_o^*, the distance is 0; otherwise, the distance is 1. And $h(\cdot)$ is a constraint function to regularize the users' weights. As users have various weights, we set the constraint function to be:

$$h(\{t_u\}) = \sum_{u \in \mathcal{U}} \exp\{-t_u\} = 1. \tag{4}$$

We choose this constraint function in order to derive meaningful user weight distribution, which is discussed later in this section.

The intuition behind this optimization formulation is as follows. Since users with high weights provide more accurate values of network quality measurement, the aggregated measurement value should be closer to the claimed values from these users. If the aggregated measurement v_o^* is far from the claimed values v_o^u from high-weight users, the loss will be high because the distance between v_o^* and v_o^u is weighted by user weight t_u. Meanwhile, we allow that the aggregated measurement v_o^* to be different from the claimed values v_o^u from users with a low weight t_u. By minimizing the overall distance, the aggregated measurement will rely on the users with high weights and their provided information.

From Eq. (3), we can see that there are two sets of unknown variables, i.e., $\{v_o^*\}$ and $\{t_u\}$. Therefore, a natural way to solve this optimization problem is to use block coordinate descent techniques [5]. Note that although block coordinate descent technique is widely adopted for optimization problems, our way of formulation (i.e., objective function and constraints) on the practical network quality measurement task is unique. Specifically, we propose a two-step procedure, which iteratively updates one set of variables to minimize the objective function while keeping the other set of variables unchanged. Here are the two steps:

- *Aggregation Step*: At this step, we fix the weight for each user and assume that $\{t_u\}$ are known. With the known user weights, we aggregate the measurement values for each status as follows:

$$v_o^* \leftarrow \arg\min_{v \in \mathcal{S}} \sum_{u \in \mathcal{U}_o} t_u \cdot dis(v_o^u, v). \tag{5}$$

  This equation is equivalent to Eq. (2). Once we calculate the aggregated measurement for each object as $\{v_o^*\}$, the overall objective function is minimized as $\{t_u\}$ are fixed.
- *Estimation Step*: At this step, we fix the aggregated value for each status and update weights for users by minimizing the following function:

$$\begin{aligned} \{t_u\} \leftarrow \arg\min_{\{t_u\}} \quad & \sum_{o \in \mathcal{O}} \sum_{u \in \mathcal{U}_o} t_u \cdot dis(v_o^u, v_o^*) \\ \text{s.t.} & \sum_{u \in \mathcal{U}} \exp\{-t_u\} = 1. \end{aligned} \tag{6}$$

  The closed-form solution for Eq (6) is:

$$t_u = -\log\left(\frac{\sum_{o:u \in \mathcal{U}_o} dis(v_o^u, v_o^*)}{\sum_{u' \in \mathcal{U}} \sum_{o:u' \in \mathcal{U}_o} dis(v_o^{u'}, v_o^*)}\right). \tag{7}$$

  According to this update equation, the weight of a particular user is the negative logarithm of normalized distance, which is measured as the distance of the user's measurement to the aggregated measurement divided by the total distance among all users. This matches our intuition that a user whose provided measurement values are closer to the aggregated value should be assigned a higher weight. The logarithm is used to re-scale the user weights to a reasonable range.

The pseudo code of this procedure is summarized in Algorithm 1. We initialize user weights uniformly, and then iteratively conduct the aggregation step and estimation step until convergence. We will give the detailed proof of convergence in the following. In practice, we implement the convergence criterion by judging whether the decrease in the objective function is small enough compared with the previous iterations. In the experiments, we find that the convergence of this approach is easy to judge because the first several iterations incur a huge decrease in the objective function, and once it converges, the results become stable.

Algorithm 1

Input: Claimed values from users for statuses $\{v_o^u | u \in \mathcal{U}_o\}_{o=1}^N$
Output: Aggregated values $\{v_o^*\}_{o=1}^N$ and user weights $\{t_u\}_{u \in \mathcal{U}}$

Initialize user weights $\{t_u = 1\}_{u \in \mathcal{U}}$;
repeat
  for $o \leftarrow 1$ to N **do**
    Aggregation step: calculate the aggregated measurement v_o^* for status o according to Eq. (5) based on the current estimation of user weights;
  end for
  for each $u \in \mathcal{U}$ **do**
    Estimation step: update users weights $\{t_u\}$ according to Eq. (7) based on the current aggregated measurement;
  end for
until Convergence criterion is satisfied
return $\{v_o^*\}_{o=1}^N$ and $\{t_u\}_{u \in \mathcal{U}}$

2.4 Time Complexity Analysis

Here we analyze the time complexity of the proposed method. Assume totally there are C claims about N objects from $|\mathcal{U}|$ users. Note that as some users may not provide observations about some objects, $C \leq N \cdot |\mathcal{U}|$. At aggregation step, weighted aggregation is conducted for each object among its corresponding claims, and thus we need $O(C)$ time to compute the aggregated measurements for all the objects. At estimation step, we need to scan all the claims to calculate the error between the claimed value and the aggregated value, so the estimation step also takes $O(C)$ time. Thus, for each iteration, the time complexity is $O(C)$. In next section, we will prove the convergence of the proposed method, which means that the number of iteration can be regarded as a constant. Therefore, the time complexity of the proposed method is $O(C)$, i.e., the running time of the proposed method is linear with respect to the number of observations. In Section 5.4, we will experimentally confirm this analysis.

2.5 Convergence Analysis

Now we prove the convergence of the proposed weighted voting approach by the following theorem.

Theorem 1. *When 0-1 loss function is used as loss function and Eq. (4) is used as constraint, the convergence of the proposed weighted voting approach is guaranteed.*

Proof. According to the proposition on the convergence of block coordinate descent [5], if unique minimum is achieved by optimizing the objective function with respect to one set of variables during each iteration, the iterative procedure leads to a stationary point. Therefore, to prove the convergence of the proposed approach for Eq. (3), we only need to show that unique minimum is achieved with respect to user weights or the aggregated values during aggregation or estimation step.

Let's first look at the aggregation step. At this step, the weights are fixed, and thus Eq. (3) is a weighted combination of all the distance between each input and the aggregated value. Clearly, v_o^* should be the value that receives the highest weighted votes among all possible values:

$$v_o^* \leftarrow \arg\min_{v \in \mathcal{S}} \sum_{u \in \mathcal{U}_o} t_u \cdot \mathbb{1}(v_o^u, v), \tag{8}$$

where $\mathbb{1}(x, y) = 1$ if $x = y$, and 0 otherwise. If we choose any other values for v_o^*, the objective function incurs a larger value.

At the estimation step, the aggregated values are fixed, and we will prove that the following solution for user weights gives unique minimum of the optimization problem Eq. (3) with constraint Eq. (4):

$$t_u = -\log\left(\frac{\sum_{o:u \in \mathcal{U}_o} dis(v_o^u, v_o^*)}{\sum_{u' \in \mathcal{U}} \sum_{o:u' \in \mathcal{U}_o} dis(v_o^{u'}, v_o^*)}\right). \tag{9}$$

Since the aggregated values are fixed, the optimization problem Eq. (3) has only one set of variables $\{t_u\}$. We first prove that Eq. (3) is convex. We introduce another variable a_u so that $a_u = \exp(-t_u)$. Now we express the optimization problem in terms of a_u:

$$\begin{aligned} \min_{\{a_u\}} \quad & \sum_{o \in \mathcal{O}} \sum_{u \in \mathcal{U}_o} -\log(a_u) \cdot dis(v_o^u, v_o^*) \\ \text{s.t.} \quad & \sum_{u \in \mathcal{U}} a_u = 1. \end{aligned} \tag{10}$$

The constraint in Eq. (10) is linear in a_u, which is affine. The objective function is a linear combination of negative logarithm functions and thus it is convex. Therefore, the optimization problem Eq. (3) with Eq. (4) is convex, and any local optimum is also global optimum [6].

We use the method of Lagrange multipliers to solve this optimization problem. The Lagrangian of Eq. (10) is given as:

$$\begin{aligned} L(\{a_u\}, \lambda) = & \sum_{o \in \mathcal{O}} \sum_{u \in \mathcal{U}_o} -\log(a_u) \cdot dis(v_o^u, v_o^*) \\ & + \lambda \cdot \left(\sum_{u \in \mathcal{U}} a_u - 1\right), \end{aligned} \tag{11}$$

where λ is a Lagrange multiplier. Let the partial derivative of Lagrangian with respect to a_u be 0, and we can get:

$$\sum_{o:u \in \mathcal{U}_o} dis(v_o^u, v_o^*) = \lambda \cdot a_u. \tag{12}$$

From the constraint that $\sum_{u \in \mathcal{U}} a_u = 1$, we can derive that

$$\lambda = \sum_{u' \in \mathcal{U}} \sum_{o:u' \in \mathcal{U}_o} dis(v_o^{u'}, v_o^*). \tag{13}$$

Plugging Eq. (13) and $t_u = -\log(a_u)$ into Eq. (12), we obtain Eq. (9). □

2.6 Practical Issues

The proposed aggregation algorithm can be used to derive an aggregated measurement value for network quality and the presentation of the algorithm is based on a specific scenario. In fact, it can be applied or easily extended to fit various scenarios and take advantages of different computing platforms. Here, we discuss some practical issues when extending the approach to these scenarios.

- *Data format*: In previous discussion, we assume that the measurement value from each individual user is only one value chosen from the predefined set $\mathcal{S} = \{\text{"}good\text{"}, \text{"}medium\text{"}, \text{"}bad\text{"}\}$. In fact, the proposed framework is not restricted to this particular set, instead, the method works well on any set. If the network provider would like to see more fine-grained predefined values, the proposed method can be easily adapted to handle the new set.
- *User weight assignment*: The constraint function assigns various distributions of weights among users. According to the requirement of application tasks, other strategies can be adopted to set the constraint function. For example, if user selection is desirable, we can force $t_u \in \{0, 1\}$, and use the corresponding constraint $h(\{t_u\}) = \frac{1}{j} \sum_{u \in \mathcal{U}} t_u = 1$. This strategy will select j users and eliminate the other users' information from the aggregation process.
- *User weight initialization*: In Algorithm 1, we initialize user weights with uniform values. However, if we have any prior knowledge or external knowledge about users, such information can be incorporated into the setting of the initial user weights.
- *Online version*: In the previous discussion, we discuss our proposed method in an offline setting, but it can be easily modified into online or incremental setting as described below. Suppose we have data continuously flowing in, i.e., data streams, and they arrive in sequential chunks. At each time stamp, based on the current estimated weights, we calculate the aggregated measurements for newly emerging statuses. Based on these aggregated measurements, for each user, we calculate their loss on these new statuses, and then we incrementally update t_u according to the following equation:

$$t_u = -\log\left(\frac{b_u + \sum_{o:u \in \mathcal{U}_o} dis(v_o^u, v_o^*)}{\sum_{u' \in \mathcal{U}} b_{u'} + \sum_{o:u' \in \mathcal{U}_o} dis(v_o^{u'}, v_o^*)}\right), \tag{14}$$

  where b_u is used to record the total loss for user u from the beginning to current, and it will be updated each time to include the new loss.
- *Parallel platform*: The number of involved users and statuses could be huge, and usually network

TABLE 1
Data Sample: User-Level Network Quality Measurements are Obtained by Combining Users' RSCP and Ec/No Information

User	Object	Measurement		Network
		RSCP (dBm)	Ec/No (dB)	Quality
u_1	o_1	−99.0000	−14.0000	bad
u_2	o_1	−93.0000	−12.5000	medium
u_3	o_1	−98.0000	−8.5000	medium
u_1	o_2	−103.0000	−12.0000	bad
u_2	o_2	−97.0000	−9.0000	medium
u_3	o_2	−83.0000	−5.5000	good
...	...	...	...	...

providers have powerful parallel computation platform. Our proposed method can take this advantage to speed up the process. Here, we briefly discuss how to fit our method into parallel computing platforms such as MapReduce. It is obvious that the aggregation step can be executed independently for each status to parallelize. The estimation step is in summation form, and according to [7], it can be parallelized by aggregating partial sums. Overall, the proposed method is easy to parallelize.

3 Data Collection

In previous section, our proposed method is presented, and several practical issues are also discussed to fit various application scenarios. In the following sections, we will describe the procedure of collecting and extracting user measurement of network quality as the input to the proposed aggregation method.

We collect three types of raw data including Measurement Report (MR) log files, IP packet traces, Performance Call History and Record (PCHR) log files, from the real Universal Mobile Telecommunication System (UMTS) network, which is one of the most popular 3G mobile cellular network deployed nowadays. Meanwhile, we maintain an IMEI (International Mobile station Equipment Identity) database which can help us to establish the mapping from the type allocation code for a device to the corresponding device type, including the hardware model and operating system. All these datasets are collected during the same period and present different features of the network. The network quality information from smartphone users will be derived from these log files, which are automatically collected by the network provider from users when they are using the network. During this process, the users get involved passively, and they do not need to take any extra effort to contribute their measurement data.

MR logs periodically record the user-level network signal quality in terms of Received Signal Code Power (RSCP) and Ec/No, which are both standard measurement in 3G network. Generally, RSCP and Ec/No represent the signal power and signal intensity received by the user in the network at that time respectively. As signal quality essentially indicates mobile cellular network quality, user-level network quality measurement can be obtained by combining these two values. According to the requirement from network provider, we map the raw values to discrete values

TABLE 2
Statistics of the Collected Datasets

	City A	City B
number of claims	34,917,421	193,480,418
number of users	135,155	481,641
number of cells	10,421	12,294

$\{``good", ``medium", ``bad"\}$ to represent network quality, as this kind of representation is easy to understand and can be further adopted for planning and scheduling. To be more specific, if a user's RSCP is smaller than threshold $-95dBm$, it indicates "bad" network quality, otherwise, it indicates "good" network quality. For Ec/No, the same strategy is used with a threshold set at $-10dB$. Note these thresholds are determined by domain experience and common practice from cellular operators. If both RSCP and Ec/No indicate "good" network quality, the combined measurement is "good"; if both of them are "bad", the combined one is "bad"; otherwise, it is "medium". By doing this, for a cell at a certain time, we map each user's RSCP and Ec/No information into "good", "medium", or "bad" as the measurement of this user about the corresponding cell he is using. Note that this mapping from raw values to discrete values is decided by the desired granularity that the network provider wants to achieve, and the proposed method can work with any predefined discrete values. A small sample of user-level information and corresponding network quality measurements are illustrated in Table 1. Besides the network quality measurements, MR logs provide the location information of each user in every record. In other words, we can get the network quality information from all the users located at certain places with respect to a cell from MR logs. We implement the proposed aggregation method on these datasets to integrate user-level network quality measurement and derive the network quality of each cell.

The IP packet traces, PCHR logs, and IMEI database provide information about the factors that are related to network quality. By correlating these three databases, we can obtain information including cell-level traffic patterns, user device types, etc. Note that the information extracted here is only used during result validation. They are not the input to the aggregation scheme.

We conduct experiments on two real network datasets collected using the above procedure. The first one is collected in City A, from November 25, 2010 to December 2, 2010. After pre-processing, it contains $34,917,421$ entries, which involve $135,155$ users and $10,421$ cells. The second one is collected in City B, from March 3, 2013 to March 6, 2013. This dataset contains $193,480,418$ entries, in which $481,641$ users and $12,294$ cells are involved. Statistics of these two datasets are summarized in Table 2. These two recently collected large-scale datasets provide a good testbed for the 3G network quality measurement task.

4 Running Example

In this section, we demonstrate how the proposed methodology works using the example shown in Table 1. In this example, we assume that three users are available and they provide network quality measurements for five statuses.

TABLE 3
Running Example: Input Data

	o_1	o_2	o_3	o_4	o_5
u_1	bad	bad	good	medium	good
u_2	medium	medium	good	good	good
u_3	medium	good	medium	bad	medium

The input from users are summarized in Table 3. The goal is to aggregate these user-input measurements to obtain more reasonable quality values for each status.

According to Algorithm 1, we start by assigning equal weights to all users, that is, we set $t_{u_1} = t_{u_2} = t_{u_3} = 1.0$, where t_{u_i} is the weight of the user u_i. Based on the initial user weights, we conduct the first round of aggregation step, which is to calculate each status's network quality by a weighted voting among user input. Currently all the users have the same weights, so the results are basically obtained by majority voting. For example, for the first status o_1, the value "medium" has 2.0 votes while "bad" only has 1.0 votes, so "medium" is selected as the aggregated measurement. All the results are illustrated in Table 4.

After the aggregation step, we update users' weights based on the current aggregated measurements. According to Eq. (7), we calculate the difference between each user' provided information and the aggregated measurements, and then re-scale it as users' new weights. In this running example, the updated user weights will be $t_{u_1} = 1.25$, $t_{u_2} = 1.25$, and $t_{u_3} = 0.84$. As can be observed from Table 4, u_3's input is usually different from the aggregated value so it receives a low weight.

Each round consists of one aggregation step and one estimation step. Table 5 shows the result of aggregation step during round 2. Similar to the first round, users' weights are re-estimated based on the new aggregated measurements, and they change to $t_{u_1} = 1.94$, $t_{u_2} = 1.25$, and $t_{u_3} = 0.55$.

At round 3, we continue to conduct aggregation step and estimation step. The result of aggregation step is shown in Table 6, and the weights will be updated as $t_{u_1} = 8.98$, $t_{u_2} = 0.98$, and $t_{u_3} = 0.47$. After this round, we will find that both the aggregated measurements and the user weights will not be changed any more. In other words, the algorithm reaches convergence, and thus Table 6 is the final output.

5 Experiments

In this section, we report the experimental results that validate the proposed crowdsourcing aggregation approach for network quality measurement. In practice, for network quality measurement task, it is very difficult to identify clearly defined quality measurement that can be adopted as the ground truth. This motivates our work but makes evaluation difficult. To tackle this challenge, we evaluate the performance of the proposed method from the following perspectives: (1) We first validate our assumption that the measurement information from users are affected by user-related factors and thus user weight assignment is needed. (2) The effectiveness of the weighted aggregation approach is justified by comparing with majority voting baseline method on simulations and case studies. (3) We then evaluate the network quality output by the proposed aggregation scheme through showing the quality evolutionary patterns with respect to different factors, such as time, data plan, etc. (4) The proposed approach is shown to be efficient on large-scale datasets and has linear running time.

TABLE 4
Running Example: Aggregation Step in Round 1

	o_1	o_2	o_3	o_4	o_5
u_1	bad	bad	good	medium	good
u_2	medium	medium	good	good	good
u_3	medium	good	medium	bad	medium
v_o^*	medium	bad	good	bad	good

TABLE 5
Running Example: Aggregation Step in Round 2

	o_1	o_2	o_3	o_4	o_5
u_1	bad	bad	good	medium	good
u_2	medium	medium	good	good	good
u_3	medium	good	medium	bad	medium
v_o^*	medium	bad	good	medium	good

5.1 User Weight Assignment

The proposed weighted aggregation approach is based on the assumption that the measurement information collected from different users have different weights due to the variance in user device, user mobility, user habit, etc. As network quality information collected from different users should not be treated equally, it is essential to estimate user weight and conduct weighted aggregation. In this part, we justify this assumption through experimental results on our data collection of 3G networks that is described in Section 3.

As we discussed, there might be multiple user-related factors that contribute to the user weight assignment. Here we choose one factor as an example to illustrate how it may affect the network quality measurements – user device. Terminal devices used by end users come from different manufacturers. These devices typically have various hardware and software systems which lead to the difference in measured signals. As some devices may present more accurate measurement values, the users who use such devices are regarded as providing better measurement information and receive high weights. Therefore, we group users based on the type of devices they are using, and we expect variability in user weights among different user groups according to device types.

From our data collection, we derive user groups based on device types as follows. We identify the type of device for each user, and find more than 500 types among all the

TABLE 6
Running Example: Aggregation Step in Round 3

	o_1	o_2	o_3	o_4	o_5
u_1	bad	bad	good	medium	good
u_2	medium	medium	good	good	good
u_3	medium	good	medium	bad	medium
v_o^*	bad	bad	good	medium	good

TABLE 7
Percentage of High-Weight Users for Different Device Type Groups

Device type	Percentage	Device type	Percentage
BlackBerry	13.56%	LG	10.89%
HTC	5.37%	Motorola	7.12%
Huawei	19.79%	Nokia	10.81%
iPad	10.28%	Samsung	12.00%
iPhone	5.87%	Sony	13.15%
Lenovo	8.52%	ZTE	17.23%

users. There exist some missing entries, so totally, we find device type information for $120,742$ out of $127,258$ users, which accounts for 95 percent. In order to visually inspect the difference in weights among user groups, we map the 500 specific device types into 12 general types according to their brands.

We then partition the whole user set into groups based on their general device types, and check the percentage of users with high weights within each group. We set a threshold and regard the users whose weight is larger than a threshold (16 in this experiment) as high-weight users. Table 7 shows the percentage of high-weight users for each device type group. From this table, we can observe the variance across different device types, which confirms the effect of user device.

To further demonstrate the difference in user weights among device type groups, we quantitatively show the pair-wise difference by t-test. Specifically, we perform t-test on the weight distributions of each pair of device type groups, which outputs whether the two distributions are significantly different or not. The results of pairwise t-test with 5 percent significant level are summarized in Table 8. Due to space limit, only the five most popular device types are shown here. The pairwise difference can be clearly observed among most of the devices. Both Tables 7 and 8 show the variation in user weights caused by device types. Other factors may also lead to the difference in the ability of users to provide measurement values. Therefore, it is critical to take into account user weights in the aggregation of measurement values to reduce the effect of user-related factors.

5.2 Quality Evaluation

In this section, we validate the effectiveness of our proposed weighted aggregation method compared with majority voting. We first evaluate both methods on a synthetic dataset in which the "ground truth" information is available, and then show the comparison on a case study from the real datasets.

TABLE 8
Pairwise t-Test on User Weights of Device Type Groups

	Nokia	iPhone	iPad	Samsung	HTC
Nokia	NA	1	0	0	1
iPhone		NA	1	1	1
iPad			NA	0	1
Samsung				NA	1
HTC					NA

1 denotes the positive t-test result while 0 denotes negative ones.

TABLE 9
Performance Comparison on Synthetic Dataset

Trial	Error Rate		Weight Distance	
	Proposed Method	Voting	Proposed Method	Voting
1	0.0825	0.1640	0.0455	0.1651
2	0.0649	0.1272	0.0469	0.1626
3	0.0614	0.1412	0.0486	0.1620
4	0.0833	0.1518	0.0444	0.1639
5	0.0789	0.1465	0.0453	0.1624
6	0.0965	0.1754	0.0444	0.1629
7	0.0851	0.1500	0.0454	0.1625
8	0.0868	0.1702	0.0486	0.1645
9	0.0789	0.1535	0.0484	0.1615
10	0.0667	0.1500	0.0446	0.1628
11	0.0877	0.1570	0.0476	0.1629
12	0.0675	0.1491	0.0460	0.1609
13	0.0904	0.1693	0.0472	0.1631
14	0.0807	0.1474	0.0451	0.1635
15	0.0982	0.1754	0.0499	0.1640
16	0.0623	0.1456	0.0459	0.1611
17	0.0667	0.1395	0.0523	0.1617
18	0.0825	0.1614	0.0481	0.1633
19	0.0746	0.1605	0.0540	0.1640
20	0.0711	0.1579	0.0440	0.1633
...	...	...	...	...
99	0.0754	0.1421	0.0430	0.1639
100	0.0737	0.1500	0.0452	0.1632
Average	0.0781	0.1541	0.0468	0.1632

5.2.1 Simulated Experiments

We randomly select 100 cells' network quality during one day (totally 100 trials), and these real trials implicitly cover various factors that can affect the network quality of cells. These real trials are regard as the "ground truth" in evaluation. Then we simulate users with different weights. For each trial, we generate measurement values of 30 users, and each user is given a parameter α ($\alpha \in [0, 1]$). A low α indicates that the corresponding user has a low weight, and his measurement values are more likely to be affected by user-related factors. On the other hand, a high α indicates a high probability of the user being unbiased and providing reliable measurement values.

According to the requirement from network provider, the network quality measurements are discrete values. Thus, majority voting, i.e., Eq. (1) is adopted as the natural baseline method. Due to the randomness of simulation, we repeat each experiment 10 times and then report its mean as the performance result.

Table 9 summarizes the results on the 100 trials of simulation. Note that due to the space limit, we cannot show all the results on 100 trials, and only the results of 22 trials and the average result of 100 trials are reported in Table 9. Error rate is used as the performance metric, which is computed as the percentage of aggregated outputs that are different from the ground truth. It is clear that the proposed method outperforms the baseline across all the trials with large margins. The baseline method treats all the users equally and assigns the same weights to them, and thus the important differences among users are not recognized in the aggregation. Different from the baseline, the proposed weighted aggregation approach considers user-related factors and estimates user weights, which results in more accurate aggregation. The

TABLE 10
Network Information for Case Study

Timestamp	Network Quality	Network Traffic (Byte)
T_{-3}	good	1,205,341
T_{-2}	good	67,210
T_{-1}	good	13,028
T_0	?	22,203
T_1	good	20,771
T_2	good	11,535
T_3	good	18,339

results demonstrate the necessity of modeling user weights in the aggregation process. We also compute the distance between the estimated user weights and the true user weights to illustrate the effectiveness of the proposed method in user weight estimation. We can see that the user weights output by the proposed method simulate true user weights well (less than 0.1 in distance to true user weights, much smaller than those of baseline).

5.2.2 Case Study

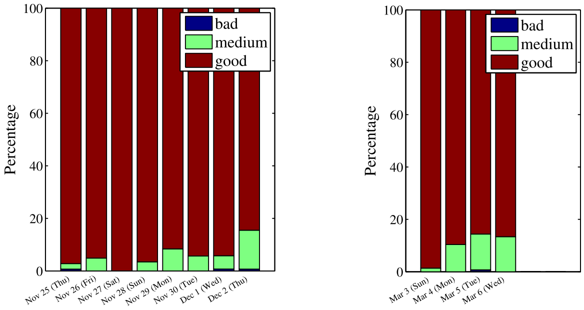
We run both the proposed weighted aggregation method and majority voting baseline on the collected 3G network datasets and we compare the obtained network quality results of these two methods. In total, 11.2 percent of the results from these two methods are different. Among them, we randomly choose one case to show how they are different and which result is more reasonable.

In this case study, we are interested in the network quality of a particular cell at a particular time T_0. Eight users provide network quality measurement values for this cell at this time. Three of them claim that the network quality is "good" while four users claim it as "medium" and one user claim it as "bad". Clearly, the majority voting baseline outputs "medium" as the aggregated value because "medium" is the majority. However, the proposed method outputs "good" because it assigns higher weights to the users that claim "good" in user weight estimation. To check which output is more reasonable, we summarize the network quality and traffic information immediately before and after T_0 to infer the quality at T_0 (shown in Table 10).

The target time stamp is T_0 as we try to detect the network quality for this time stamp. From Table 10, we can observe that before and after T_0, the network quality is "good". Network traffic does not show significant increases at T_0 compared with other time stamps. Therefore, it is reasonable to assume that network quality at T_0 should be similar to that at the other time, which is "good". This justifies the output of the proposed weighted aggregation method. Compared with majority voting that treats all users equally, our proposed approach is able to obtain accurate estimation of user weights by considering the effect of user-related factors, and thus can output more reasonable aggregated measurement.

5.3 Network Quality Patterns

We conduct the user weight estimation and weighted aggregation on the network quality measurement data collected from users. The algorithm outputs an aggregated network measurement value for each cell at each time. As we do not have the ground truth network quality, we conduct case studies to show the evolution of aggregated network quality values for some selected cells as time elapses. We find that cells' network quality changes in various ways, and each of them is the result of various reasons: weekdays or weekends, locations, users' 3G data plans, weather conditions, etc. The evolution of network quality for different cells are different as each cell may have different sets of users and environmental factors, but we identify some common patterns that indicate the effect of factors. Next we show some of these patterns and provide meaningful explanations by checking the possible factors that may cause the changes in network quality.



(a) A cell in city A (b) A cell in city B

Fig. 1. Effect of weekdays on cells' network quality.

In the following case studies, we focus on explaining the network quality patterns from the perspective of network usage. We analyze several factors that may affect the usage of the network and in turn affect network quality. Typically, a large amount of users and heavy usage of networks may result in the quality degrade while a small number of users and light usage contribute to quality improvement. To further demonstrate this connection, we extract cell-level network traffic information by correlating IP packet traces, PCHR logs and IMEI database, and show the relationship between network traffic and network quality.

Weekdays or weekends. In downtown or residence areas, people exhibit very different network usage behavior during weekdays and weekends. Fig. 1 shows the network quality evolution during a week for some cells in downtown with many companies in the area, where x-axis is the date and y-axis shows the percentage of "bad", "medium" and "good" network quality during the particular day. As can be seen, in weekdays the percentage of "bad" and "medium" network quality is larger than the ones in weekend. For example, in Fig. 1b, we observe that the network quality on Sunday is maintained as "good" 99 percent of the time, while on weekdays, the network quality drops to "medium" or "bad" cases much more often (around 11 percent of the time in a day). This is a reasonable estimate as during weekdays many users go to work in this area, and a heavy usage of network leads to the degrade in network quality. On the other hand, during weekends (third day and fourth day in Fig. 1a, first day in Fig. 1b), network quality is good as most of the people stay at home.

To confirm the reasoning about this pattern, Fig. 2 plots the comparison between network quality and traffic for the same cell in Fig. 1a. In Fig. 2, x-axis is the time (by hour) for a particular day, and y-axis is the network quality or traffic during that hour. Showing all the days may need too much space, thus we select one day for weekday and one day for

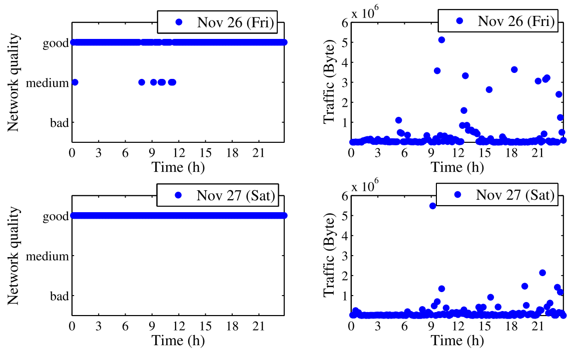
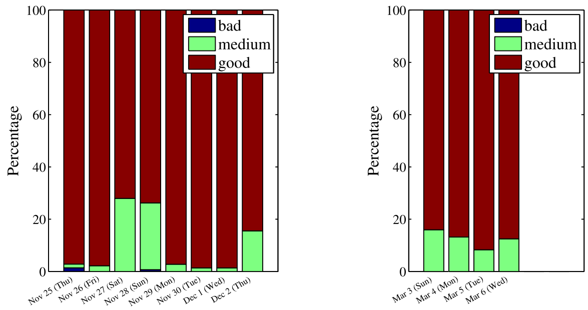
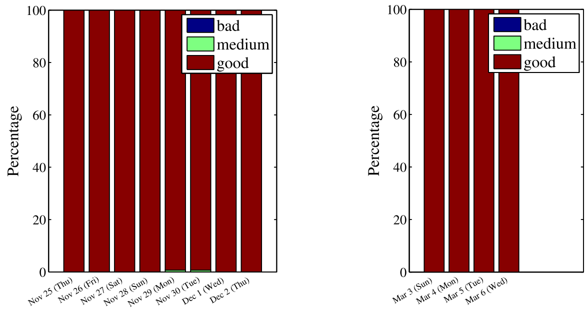


Fig. 2. Network quality and corresponding traffic for Fig. 1a.



(a) A cell in city A (b) A cell in city B

Fig. 3. Effect of weekends on cells' network quality.



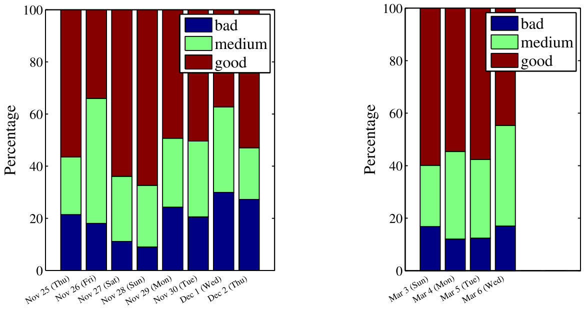
(a) A cell in city A (b) A cell in city B

Fig. 4. Effect of locations on cells' network quality.

weekend to illustrate the comparison. It is clear that weekdays witness traffic burst compared with weekends, which supports our claim that networks experience heavier usage and lower quality during weekdays in this area as discussed above. Please note that for some timestamps, traffic burst occurs while the network quality remains good, which is caused by the fact that the number of users within those timestamps is relatively small. This indicates that the network quality is affected by network traffic, but the network quality is not purely determined by the traffic only.

In Fig. 3, we present patterns in network quality for residential areas. In contrast to the downtown locations, the network quality in residential areas decreases during weekends (Third day and fourth day in Fig. 3a, first day in Fig. 3b) as well as weekday evenings (eighth day in Fig. 3a and the rest days in Fig. 3b) due to heavier usages. During business hours, the network quality is better as people go to their working places.

These patterns we find in network quality with respect to weekdays and weekends are commonly observed and repeat themselves periodically. Such patterns capture users' everyday mobility activities and can be used to predict network usage behavior. Network providers can accordingly adjust network parameters to provide network services of better quality.



(a) A cell in city A (b) A cell in city B

Fig. 5. Effect of locations on cells' network quality.

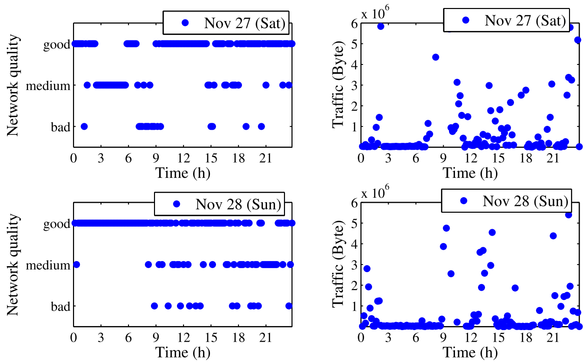
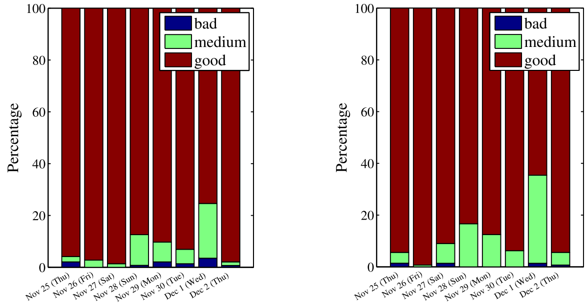


Fig. 6. Network quality and corresponding traffic for Fig. 5a.

Locations. Figs. 4 and 5 show that there exist cells with constantly high or low network quality. In Fig. 4, the cells have good quality all the time, while the cells shown in Fig. 5 always exhibit poor network quality. We identify the locations of these cells and find that places with few people and lighter usage loads typically have high-quality networks but places with constantly high population and frequent network usage have low-quality networks. The locations shown in Fig. 4 are isolated places near airport or highway, while the locations of Fig. 5 are popular places such as schools and attractions. Further, Fig. 6 shows the network traffic of a cell located at a popular attraction. From this figure, we observe that network traffic keeps at a high level most of the time so it is expected to have low network quality.

These patterns reveal the variation in base station loads, thus can be useful for better scheduling in base station construction. With these observed patterns in network quality, network providers can have a better idea on where they should build more base stations to improve the network quality of that area.

Data plan. In Fig. 7, we present an interesting pattern related to data plan usage. We can see that on December 1st, the percentage of "medium" and "bad" network quality are between 20 and 30 percent while for the rest days it is below 10 percent. This is in fact caused by the policy of users' data plans. In the collected datasets, the billing cycle is always the first day of each month. A data plan has volume limit each month, and users typically have very limited



(a) A cell in city A (b) A cell in city A

Fig. 7. Effect of data plans on cells' network quality.

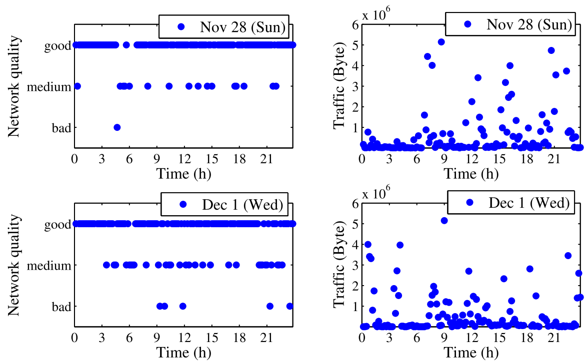


Fig. 8. Network quality and corresponding traffic for Fig. 7a.

quota left at the end of the month. To avoid usage overflow, they tend to decrease their usage of 3G network work. When a new month starts, users get sufficient data quota, and start to surf the web immediately. This causes the network usage go up and network quality go down during the first day of a month.

In Fig. 8, we show one cell's network quality and corresponding traffic on December 1st. As a comparison, we also plot this cell's network quality and traffic for another day. It is obvious that on December 1st, heavy usage occurs at non-peak hours (e.g., the midnight on this day), which validates the above explanation towards this interesting pattern.

This pattern shows the effect of data plan policy on network quality. Network providers may consider to modify existing policy to promote balanced usage among each month. For example, promotion offers can be made to encourage users to use 3G networks even at the end of their monthly plan, or users' plan can start at any day in a month to avoid competition for network resources at the beginning of a month.

Weather conditions. Another factor that causes changes in network quality is the condition of weather. The effect of weather condition is obvious: Network quality of certain cells drops when weather becomes bad. Table 11 shows the weather conditions during eight days and Fig. 9 plots the

TABLE 11
Weather Conditions for City A During Eight Days

Day	Weather Condition	Day	Weather Condition
Nov 25	clear	Nov 29	clear
Nov 26	clear	Nov 30	clear
Nov 27	clear	Dec 1	fog
Nov 28	clear	Dec 2	rain

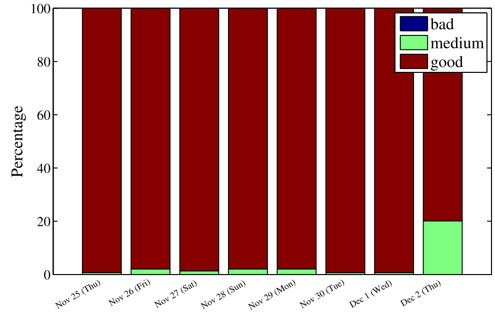


Fig. 9. Effect of weather conditions on a cell's network quality.

network quality changes of a cell in the data collection of city A. As an example, we find that in the morning of eighth day in the data collection of City A, it rains heavily so the network quality of that day is bad compared with the other days. With real-time network quality monitoring, network providers can recognize such problems and react timely to fix problems or issues that lead to the performance drop in network services.

In this section, we explain the aggregated quality measure we obtain using our proposed crowdsourcing aggregation approach by analyzing the connection of aggregated network quality to several environmental factors. Note that the cases we show may have one or two outstanding factors, but still network quality is influenced by multiple factors jointly. Therefore, instead of modeling these factors one by one, the best practice is to deploy a network quality monitoring system that can provide network providers accurate network quality measurements to help them adjust network services. The proposed aggregation framework will be a key component of this quality monitoring system because it provides robust and accurate estimates of network quality in a cost-effective way.

5.4 Running Time

In this part, we show the efficiency and scalability of the proposed approach. In Section 2.4, the time complexity of the proposed approach is analyzed, and it shows that the running time of the proposed method is linear respect to the number of claims. To experimentally demonstrate this result, we sample different number of claims from both of the two real datasets and record the running time of the proposed approach. Further, the running time of majority voting is also reported, which has the optimal efficiency. From Fig. 10, we can observe that although the proposed method requires

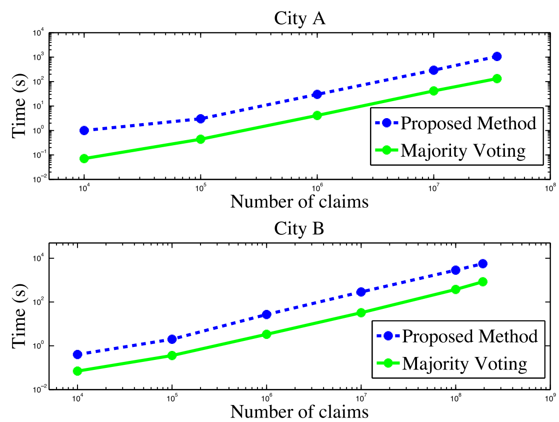


Fig. 10. Running time on collected datasets.

TABLE 12
Running Time Comparison on Large-Scale Datasets

#claim	Single-Machine	MapReduce
1,000,000	27 s	100 s
10,000,000	288 s	192 s
100,000,000	2860 s	669 s
193,480,418	5633 s	984 s

more computation time than majority voting, both of them have linear complexity with respect to the number of claims.

In Section 2.6, we mention that the proposed method can be deployed on parallel system, which enables the proposed method to deal with large-scale data efficiently. To demonstrate this advantage, we implement the parallel version of the proposed method based on Hadoop, and repeat the above experiments on a 15-node Dell Hadoop cluster with Intel Xeon E5-2403 processor. For the purpose of comparison, we report the running time of both the single-machine version and MapReduce version of the proposed method in Table 12. When the number of claims is small, the parallel version of the proposed method requires a bit more computation time due to the communication cost among parallel system. However, when the number of claims increases, the benefit of the parallel version becomes significant. These observations demonstrate that the proposed method can be applied to network services in real practice to facilitate real-time network quality monitoring.

5.5 Summary of Insights

We proposed a novel crowdsourcing approach to overcome the limitations of traditional drive testing approach so that network providers can effectively monitor network quality. The proposed approach is based on a weighted aggregation of user-provided measurements where users are weighted according to their user-related factors. This is motivated by the fact that user-provided network information could be affected by user device, user mobility, user habit or other user-related factors. In Section 5.1 we validate this assumption by checking the user weight assignment and its relationship with device types.

We show the effectiveness of the proposed approach by conducting quantitative analysis in Section 5.2. First, the evaluation is conducted on synthetic datasets, in which the ground truth information is available to conduct quantitative analysis. The error rate of the proposed methods is only around a half of the error incurred by the baseline method. Then we show how and why the proposed weighted aggregation method outperforms baseline method through real case study.

Then we apply the proposed crowdsourcing approach on two large-scale real-world datasets involving 616,796 users and 22,715 cells. Based on the aggregated results, we show some network quality evolutionary patterns in Section 5.3. From these patterns, we can clearly identify the impact of weekdays and weekends, user data plans, etc. on network quality. These patterns are further validated by comparing network quality and the corresponding traffic. According to these patterns, network providers can adjust cell access control, scheduling and planning to avoid quality decrease and provide better services to network users.

Last but not the least, we demonstrate that the proposed method runs in linear time with respect to the size of the dataset, and we also implement the parallel version of the proposed method based on Hadoop. This shows that the proposed method has good scalability and is suitable for real-time quality monitoring.

6 Related Work

There are two types of network quality evaluation methods to infer users' experience: Objective and subjective methods. This partition is based on the parameters or data used to estimate the network quality: Subjective methods require users' personal experience reports as input while objective methods use various network measurements.

An example of subjective methods is [2], in which the authors proposed a framework to measure network quality in a subjective way: Users can click a button on the interface when they are unhappy about the current network quality. In practice, as subjective methods need the cooperation from users, it may not be able to be deployed in large scale easily.

For objective methods, various parameters and performance metrics, such as network traffic and speech exchange rate, can be used to derive network quality. Among them, two typical places to extract information for network quality inference are from data-plane and control-plane. In data-plane performance analysis, the traffic of GPRS/UMTS network is characterized by TCP performance [8] and round-trip time of TCP flow data [9]. In [10], the similarities and differences between the traffic of CMA2000 network and wireline data traffic are presented. Furthermore, the traffic dynamics of cellular devices are characterized according to device types and application dimensions in [11]. As for control-plane performance analysis, in [12], data packet headers and various signaling messages are collected from a national 3G network, and used to analyze their temporal and spatial variations. In [13], the authors analyze the RRC state transitions of user sessions using collected traces.

The main difference between the above related work and our work is: The related work studies the network quality measurement for individual users, while our work is in an orthogonal perspective that we measure the network quality for network providers. We propose a weighted crowdsourcing approach by aggregating individual users' information.

The proposed approach estimates user weights to capture user-related factors. Some existing work recognizes the diversity of user behaviors, and thus motivates and supports our method to estimate user weights. For example, several papers investigate the variance and diversity in smartphone usage as follows. In [14], the authors presented a detailed analysis of traffic on smartphones with 43 users. Later, they found the diversity in user behavior by analyzing the dataset collected from 225 individual users [15]. [16] identifies diverse usage behavior of smartphones from the perspective of apps on national-level networks. In [17], the authors studied the geospatial correlations between network traffic and application usage from user perspective. Note that although the diversity among users has been recognized, such information is not taken into account in network quality evaluation previously. As far as we know, this is the first work to characterize user-related factors and take such valuable information in the task of network quality measurement aggregation.

Recently, [18] studied the phenomenon that during crowded events, cellular network quality degrades. Based on their findings, two mechanisms were suggested to improve network quality. It is interesting to see that one of our observed patterns about users' data plan (Fig. 7) can be considered as a crowded event, which is consistent with their observation.

In [19], the authors studied the effect of network protocol and application behavior on the performance of 4G LTE network. Although we use a 3G network dataset in this paper, the proposed method can be adapted to 4G networks, which is considered to be future work. We also plan to explore the aggregation of other metrics for network quality, such as throughput, loss rate, or other contexts of data [20].

7 Conclusion

Network quality is a very important metric for network providers to consider when making scheduling and planning decisions to ensure satisfactory user experience. The traditional drive testing approach to measure network quality suffers from the limitations of high cost, limited coverage, and out-of-date measurement values. In this paper, we propose a novel strategy to report up-to-date and accurate estimates of network quality with little costs. The basic idea is to collect measurement values from a large amount of users who participate in the networks and aggregate the measurement values to obtain consensus quality measurement. We also propose a weighted aggregation method to capture the fact that user-provided measurement values are affected by user-related factors. The proposed aggregation method estimates user weights and aggregated network quality measurement simultaneously using a joint optimization framework. The solution to the optimization problem improves the user weight and network quality estimates iteratively. Experimental results on two real datasets collected from two cities' 3G networks demonstrate the effectiveness and efficiency of the proposed aggregation strategy. As shown in the results, network quality patterns are affected by several possible environmental factors. The proposed method outperforms majority voting baseline in its ability of better characterizing network quality and it is efficient with linear time complexity.

References

[1] I. Kostanic, N. Mijatovic, and S. D. Vest, "Measurement based qos comparison of cellular communication networks," in *Proc. IEEE Int. Workshop Tech. Committee Commun. Quality Rel.*, 2009, pp. 1–5.

[2] K.-T. Chen, C.-C. Tu, and W.-C. Xiao, "Oneclick: A framework for measuring network quality of experience," in *Proc. IEEE Int. Conf. Comput. Commun.*, 2009, pp. 702–710.

[3] K.-T. Chen, C.-Y. Huang, P. Huang, and C.-L. Lei, "Quantifying skype user satisfaction," in *Proc. Conf. Appl., Technol., Archit., Protocols Comput. Commun.*, 2006, pp. 399–410.

[4] Q. Li, Y. Li, J. Gao, B. Zhao, W. Fan, and J. Han, "Resolving conflicts in heterogeneous data by truth discovery and source reliability estimation," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2014, pp. 1187–1198.

[5] D. P. Bertsekas, *Non-Linear Programming*, 2nd ed. Belmont, MA, USA: Athena Scientific, 1999.

[6] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[7] C.-T. Chu, S. K. Kim, Y.-A. Lin, Y. Yu, G. Bradski, A. Y. Ng, and K. Olukotun, "Map-reduce for machine learning on multicore," in *Proc. Adv. Neural Inf. Process. Syst.*, 2006, pp. 281–288.

[8] F. Ricciato, F. Vacirca, and M. Karner, "Bottleneck detection in UMTS via TCP passive monitoring: A real case," in *Proc. ACM Conf. Emerging Netw. Exp. Technol.*, 2005, pp. 211–219.

[9] J. Kilpi and P. Lassila, "Micro-and macroscopic analysis of RTT variability in GPRS and UMTS networks," in *Proc. 5th Int. IFIP-TC6 Conf. Netw. Technol. Serv. Protocols Perform. Comput. Commun. Netw. Mobile Wireless Commun. Syst.*., 2006, pp. 1176–1181.

[10] J. Ridoux, A. Nucci, and D. Veitch, "Seeing the difference in IP traffic: Wireless versus wireline," in *Proc. IEEE Int. Conf. Comput. Commun.*, 2006, pp. 1–12.

[11] M. Z. Shafiq, L. Ji, A. X. Liu, and J. Wang, "Characterizing and modeling internet traffic dynamics of cellular devices," in *Proc. ACM SIGMETRICS/Int. Conf. Meas. Model. Comput. Syst.*, 2011, pp. 305–316.

[12] U. Paul, A. P. Subramanian, M. M. Buddhikot, and S. R. Das, "Understanding traffic dynamics in cellular data networks," in *Proc. IEEE Int. Conf. Comput. Commun.*, 2011, pp. 882–890.

[13] F. Qian, Z. Wang, A. Gerber, Z. M. Mao, S. Sen, and O. Spatscheck, "Characterizing radio resource allocation for 3G networks," in *Proc. ACM SIGCOMM Conf. Internet Meas.*, 2010, pp. 137–150.

[14] H. Falaki, D. Lymberopoulos, R. Mahajan, S. Kandula, and D. Estrin, "A first look at traffic on smartphones," in *Proc. ACM SIGCOMM Conf. Internet Meas.*, 2010, pp. 281–287.

[15] H. Falaki, R. Mahajan, S. Kandula, D. Lymberopoulos, R. Govindan, and D. Estrin, "Diversity in smartphone usage," in *Proc. 8th Int. Conf. Mobile Syst., Appl. Serv.*, 2010, pp. 179–194.

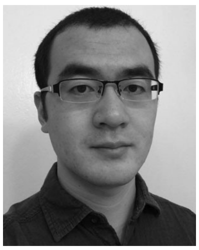
[16] Q. Xu, J. Erman, A. Gerber, Z. Mao, J. Pang, and S. Venkataraman, "Identifying diverse usage behaviors of smartphone apps," in *Proc. ACM SIGCOMM Conf. Internet Meas.*, 2011, pp. 329–344.

[17] M. Z. Shafiq, L. Ji, A. X. Liu, J. Pang, and J. Wang, "Characterizing geospatial dynamics of application usage in a 3G cellular data network," in *Proc. IEEE Int. Conf. Comput. Commun.*, 2012, pp. 1341–1349.

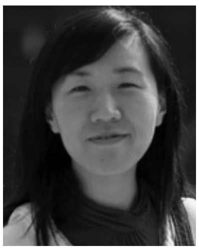
[18] M. Z. Shafiq, L. Ji, A. X. Liu, J. Pang, S. Venkataraman, and J. Wang, "A first look at cellular network performance during crowded events," in *Proc. ACM SIGMETRICS/Int. Conf. Meas. Model. Comput. Syst.*, 2013, pp. 17–28.

[19] J. Huang, F. Qian, Y. Guo, Y. Zhou, Q. Xu, Z. M. Mao, S. Sen, and O. Spatscheck, "An in-depth study of LTE: Effect of network protocol and application behavior on performance," in *Proc. ACM Conf. SIGCOMM*, 2013, pp. 363–374.

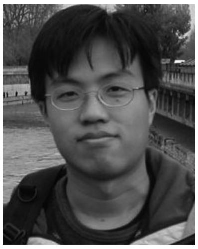
[20] X. He, P. P. Lee, L. Pan, C. He, and J. C. Lui, "A panoramic view of 3G data/control-plane traffic: Mobile device perspective," in *Proc. 11th Int. IFIP TC 6 Conf. Netw.*, 2012, pp. 318–330.



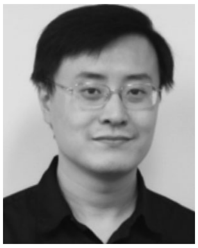
Yaliang Li received the BS degree from Nanjing University of Posts and Telecommunications in 2010. He is currently working toward the PhD degree in the Department of Computer Science and Engineering, SUNY Buffalo. His research topics include truth discovery, text and web mining, privacy-preserving data mining, and data mining application in healthcare.



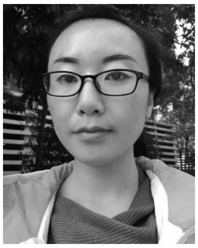
Jing Gao received the PhD degree from Computer Science Department, University of Illinois at Urbana-Champaign in 2011, and subsequently joined SUNY Buffalo in 2012. She is an assistant professor in the Department of Computer Science and Engineering at SUNY Buffalo. Her research interest include data and information analysis with a focus on truth discovery, crowdsourcing, information integration, ensemble methods, mining data streams, transfer learning, and anomaly detection. She is a member of the IEEE.



Patrick P. C. Lee received the BEng degree (first-class Hons.) in information engineering from the Chinese University of Hong Kong in 2001, the MPhil degree in computer science and engineering from the Chinese University of Hong Kong in 2003, and the PhD degree in computer science from Columbia University in 2008. He is currently an associate professor in the Department of Computer Science and Engineering, Chinese University of Hong Kong. His research interests include various applied/systems topics including storage systems, distributed systems and networks, operating systems, dependability, and security. He is a member of the IEEE.



Lu Su received MS degree in statistics and the PhD degree in computer science both from University of Illinois at Urbana-Champaign, in 2012 and 2013, respectively. He is an assistant professor in the Department of Computer Science and Engineering at SUNY Buffalo. His research focuses on the general areas of cyber-physical systems, wireless and sensor networks, and mobile computing. He has also worked at IBM T. J. Watson Research Center and National Center for Supercomputing Applications. He is a member of the ACM and IEEE.



Caifeng He received the bachelor's and master's degrees in computational mathematics both from Jilin University in 2009 and 2011, respectively. Her research interests include traffic analysis, fault demarcation, users profiling in mobile communication networks and streaming machine learning algorithms for supporting on-line applications in mobile networks.

Cheng He received the BS and MS degrees in communication and information systems from UESTC, Chengdu, in 2003 and 2006, respectively. He joined Huawei Research Institute in 2006, and now he is a principal engineer and research manager in Noahs Ark Lab of Huawei. His research interests include traffic measurement & modeling, distributed stream computing, and big data stream mining & online learning for intelligent network management and control.

Fan Yang received the BS and MS degrees in electrical engineering from XiDian University, in 2007 and 2010, respectively. His research interests include telecom network optimization, network traffic measurement and modeling, and internet financial data modeling.

Wei Fan received the PhD degree in computer science from Columbia University in 2001. He is currently the senior director and the deputy head of Baidu Big Data Lab in Sunnyvale, CA. His main research interests and experiences are in various areas of data mining and database systems, such as, deep learning, stream computing, high performance computing, extremely skewed distribution, cost-sensitive learning, risk analysis, ensemble methods, easy-to-use nonparametric methods, graph mining, predictive feature discovery, feature selection, sample selection bias, transfer learning, time series analysis, bioinformatics, social network analysis, novel applications, and commercial data mining systems. His co-authored papers received ICDM'06/KDD11/KDD12/KDD13/KDD97 Best Paper & Best Paper Runner-up Awards. He led the team that used his Random Decision Tree (www.dice.com) method to win the 2008 ICDM Data Mining Cup Championship. He received the 2010 IBM Outstanding Technical Achievement Award for his contribution to IBM Infosphere Streams. He is an associate editor of the *ACM Transactions on Knowledge Discovery and Data Mining*. During his time as the associate director of the Huawei Noah's Ark Lab in Hong Kong from August 2012 to December 2014, he led his colleagues to develop Huawei StreamSMART which is a streaming platform for online and real-time processing, query, and mining of very fast streaming data. StreamSMART is three to five times faster than STORM and 10 times faster than SparkStreaming, and was used in Beijing Telecom, Saudi Arabia STC, Norway Telenor, and a few other mobile carriers in Asia. Since joining Baidu Big Data Lab, he has been working on medical and healthcare research and applications, such as deep learning-based disease diagnosis based on NLP input as well as medical dialogue robot.