Characterizing and predicting mobile application usage

Keun-Woo Lim, Stefano Secci, Lionel Tabourier, Badis Tebbani

Characterizing and predicting mobile application usage

Keun-Woo Lim[a], Stefano Secci[a], Lionel Tabourier[a], Badis Tebbani[b]

[a] *Sorbonne Universités, UPMC Univ Paris 06, CNRS, LIP6 UMR 7606, 4 place Jussieu 75005 Paris (e-mails: keunwoo.lim@upmc.fr, stefano.secci@upmc.fr, lionel.tabourier@upmc.fr)*

[b] *UCOPIA Communications, 92120 Montrouge, France (e-mail: badis.tebbani@ucopia.com)*

Abstract

In this paper, we propose data clustering techniques to predict temporal characteristics of data consumption behavior of different mobile applications via wireless communications. While most of the research on mobile data analytics focuses on the analysis of call data records and mobility traces, our analysis concentrates on mobile application usages, to characterize them and predict their behavior. We exploit mobile application usage logs provided by a Wi-Fi local area network service provider to characterize temporal behavior of mobile applications. More specifically, we generate daily profiles of "what" types of mobile applications users access and "when" users access them. From these profiles, we create usage classes of mobile applications via aggregation of similar profiles depending on data consumption rate, using three clustering techniques that we compare. Furthermore, we show that we can utilize these classes to analyze and predict future usages of each mobile application through progressive comparison using distance and similarity comparison techniques. Finally, we also detect and exploit outlying behavior in application usage profiles and discuss methods to efficiently predict them.

Keywords: Data analytics, Clustering, Data consumption behavior, Mobile applications

1. Introduction

The worldwide dissemination of portable smartphones has completely changed the data consumption behavior of humans during the past decade. Indeed, data consumption has dramatically increased in volume and frequency, types of data have become more diversified, and a majority of users

access such data using battery-powered mobile embedded devices. To cope with this revolution, mobile operators have long sought to provide faster mobile networking technologies, resulting in breakthrough technologies such as 4G/LTE. However, considering the expected increase in mobile data usage as foreseen by Cisco's VNI index [1], even such an impressive technological progress is not considered to be sufficient for the near future, prompting various industries and institutes to examine into rapidly realizing 5G [2].

One of the most prominent methods for providing better communication service is to have a better understanding of where, when, and how people use these communication technologies. As a result, research on characterizing data consumption behavior from mobile communications has recently gained much interest. As already explored by various researchers, we can exploit usage information from various mobile service providers to discover interesting phenomena related to individual human mobility, social interactions, application usage, etc. In particular, profiling various aspects of human behavior has been a prominent area of research, as being able to predict the behavior of masses of individuals allows many different proactive planning and service adaptation to suit the needs of users in the future.

While existing research shows that call data records (CDR) information has proven to be useful for mobility pattern characterization [3] [4], it is rather limited in providing a rich characterization of mobile users behavior, and in particular in relation to their mobile Internet data consumption behavior. Indeed, nowadays mobile Internet traffic takes a large piece of mobile access network resources, and is expected to overcome mobile call traffic volumes. As outlined in [5], the current interest in mobile network data analytics is shifting toward the characterization of both user mobility and mobile data consumption. There are many application domains that can benefit from a better understanding of mobile data consumption. With the expected integration of network functions virtualization (NFV) [6] and mobile edge computing (MEC) [7] systems in forthcoming 5G infrastructure, eventually coupled with programmable network interfaces and equipment and centralized control, novel needs are expressed to grant flexibility in network and service management. For instance, being able to predict with an acceptable accuracy a sudden crowd effect in data consumption on a per-application way can support adaptive scale in/out of virtual servers in NFV and MEC infrastructures and even legacy cloud infrastructure. Dually, virtual link Quality of Service (QoS) reservation operations at the controller level of mobile backhauling network [8] [9] [10] can be triggered as a result of mobile

application usage estimation, in particular if explicit traffic engineering and software-defined network interfaces exist.

In this paper, we propose a method of clustering data consumption behavior in terms of "what" types of services users use at specific times. The main difference between our work using Wi-Fi cloud data and other works based on cellular mobile data, is that our dataset corresponds to a local scale network (e.g., restaurants, malls etc.), whereas datasets from cellular mobile networks focus on larger aggregation scales (wide area, metropolitan, nationwide) – see [11] and [5] for surveys. Consequently, we expect that our dataset can capture other kinds of mobile usage behavior. Furthermore, this dataset allows us to investigate the data usage itself. We can analyze traffic patterns of different web services, while most works focus on interactions between users (calls, text messages, contacts etc.).

After describing the related work on mobile data analysis in Section 2, we explain in Section 3 the dataset provided by a rapidly growing Wi-Fi cloud access provider in France. From this dataset, we specifically focus on extraction of data traffic usage per application at a wireless local area network (WLAN) scale, which gives us fine-grained information of how users behave at this local scale. Using this dataset, we propose a lightweight methodology in Section 4 that analyzes mobile data usage logs of anonymized users connecting to the Wi-Fi cloud. It allows us to analyze when and how people connect and consume different types of data from using mobile applications. Specifically, we create daily "profiles" of each application usage that are aggregated with each other to form multiple classes that have different characteristics. For profile aggregation, we sort the daily profiles according to the daily usage patterns and differentiate them, creating distinct classes through similarity comparison techniques. Two different types of clustering is considered: homogeneous clustering where equal number of profiles are used to create evenly balanced classes, and heterogeneous clustering where the number of profiles to create each class can be different depending on their pattern. In Section 5, we show through extensive analysis that our method can clearly distinguish a number of classes that can be utilized to predict future usage of each application. Then, in Section 6, we investigate detection of outlying behavior for each application. Using the aforementioned clustering methods, we can derive outlying daily profiles that do not behave according to one of the classes. We show that this method can be used to identify outliers in on-demand basis, and we assess their effect on the performance of our prediction technique. We discuss some future issues and conclude our

work in Sections 7.

2. Related work

One of the research areas in exploitation of mobile data logs are based on characterizing the nature of human mobility. Work such as [3] provides an analysis of tracking the mobility of phone users to show that humans do not have random trajectories but have both temporal and spatial regularities. Further researches by [12] and [13] experiment with data from different scales and datasets from various metropolitan areas, and show that different cities can have variances in human mobility patterns. A recent work by [4] showed that call data records can be organized into profiles and clustered using the spatiotemporal usage characteristics of each profile, allowing for accurate prediction and possible adaptation of the network according to the usage dynamics. However, despite these interesting results, there have also been concerns that call data records are biased [14] in the sense that data records are always limited and artificially chosen, and that it may not be a highly accurate method in characterizing general human behavior. Also, mobile user information such as CDR is only so informative as to analyze limited user behavior characteristics such as mobility and crowd information.

It is important to note, as already stated by [15], that usage of smartphones is much more diversified beyond the point of phone calls and mobile tracking. Various researchers have made some analysis regarding usage of applications and data using smartphones such as [16] and [17]. These works are different from existing work in the sense that they collect the logs from the user's smartphone and analyze the application usage behavior of individual users. On the other hand, [18] and [19] make their analysis of smartphone apps usage using anonymized datasets from Internet service providers and cellular networks, which are more diversified and abundant. [18] analyzes correlations between various parameters such as data volume, subscribers, access time, and various applications; their analysis is helpful in increasing QoS of the providers. Even though the datasets used in these researches have some similarities, our work differentiates from existing literature in the sense that we focus on: 1) profiling and clustering of different mobile applications using temporal characteristics, and 2) predicting usage of an individual application at a more fine-grained scale.

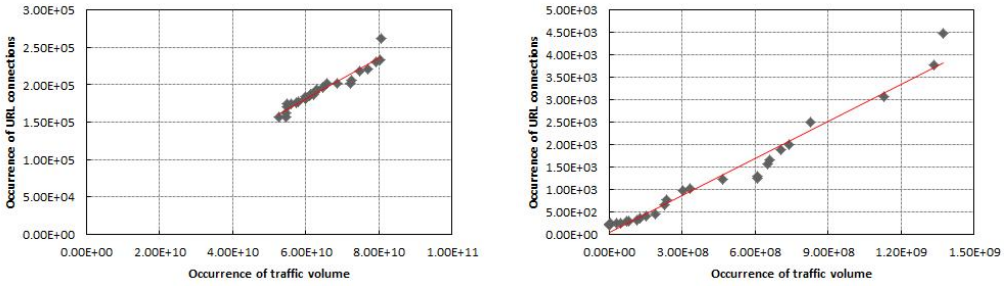
3. Dataset description

Before explaining our methodology for clustering, we give a description of the dataset that we used. We utilize the mobile application usage logs collected from a Wi-Fi cloud service provided by a rapidly growing operator located in France. The data are collected from Wi-Fi cloud locations where multiple access points exist. The location of the Wi-Fi cloud is a public area where every people can freely access. The collected logs include a session log, which contains information of all connection sessions initialized by the users during the time period. Each session log records the start and end time of the session, device information, and the data volume used during the session. From the provided session logs, we specifically analyze the session logs that were generated in four months from March to June, 2014, for closer examination and clustering. About 2500 to 3500 users access the network each month, with about 80% connected with smartphones, while the other 20% are connections using laptops and tablet computers. The whole dataset consists of 60 million URL connection logs generated from TCP traffic passing through port 80 and port 8080. This accounted for about 1858 gigabytes of ingoing/outgoing data per month, or about 60 gigabytes per day.

We note that the network controller of such Wifi networks cannot capture complete URLs in HTTPS connections, as the URL is encrypted in such connections [20]. However, in a limited manner, partial information of the URL such as the domain name can be acquired also for HTTPS sessions. One such approach is to exploit the domain name system (DNS), where the controller can log the DNS request containing the host name and the IP address response from DNS, which happens for DNS resolutions not locally cached at the client. Another approach is to exploit the server name indication (SNI) [21] during the transport layer security (TLS) signaling phase, where the *extension_data* field in the client hello messages contains the hostname. These methods cannot retrieve full URLs from each packet, but only the domain name part of an URL from preliminary signaling packets, which matches our needs as we use truncated URL information.

Note that the URL information is anonymized in the paper due to privacy restrictions. These URL connection logs are references to the sites that mobile applications access, which denotes the number of connections. We measure the number of connections to each mobile application. In this dataset, we do not have access to the data volume usage for each application at each timestep, which would be a better measurement of the actual

data consumption. However, we argue that the number of connections and data volume are highly correlated, hence, we make the assumption that our analysis could be done in the exact same way with data volume usage. In support of this idea, we show in Fig. 1 that the number of connections and the data volume are strongly correlated when compared with each other in longer periods of time (i.e., usage per hour in one day, usage per day in one month). Fig. 1(a) shows the qq-plot of the trend of monthly data usage rate (in March) as a function of the number of connections, showing that these quantities are roughly proportional. On the other hand, Fig. 1(b) plots a daily example of correlation between the two parameters, with a high level of correlation.



(a) Monthly usage correlation

(b) Daily usage correlation

Figure 1: Correlation of data volume and number of connections

It is worth noting that the association of a URL to a mobile application is not straightforward. Indeed, a URL can be associated to an application server, we have to notice that (i) a single mobile application could use multiple URLs concurrently, hence multiple application servers, and (ii) an application server can be used by multiple applications. We use truncated URLs, which hide the confidential information that may exist in the latter parts of the URL, while the domain and hostname are visible (possibly also including 2nd-level domains for short domain names). We compared these truncated URLs in order to identify the ones referring to the same mobile applications. (i.e. identical domain and hostname). This preprocessing demands to restrict ourselves to the most frequent truncated URLs. By this way, we restrict the bias due to both (i) and (ii) to a level that we conjecture as being acceptable. Therefore, the expression 'mobile application' used in

the following is a simplification referring to this disambiguation process.

From the data logs, we specifically acquire connection information of 20 most popular mobile applications that users connected to during the four-month period. These top 20 applications are defined as the 20 most visited mobile applications in terms of number of connections during the considered time period. Here, we utilize only the top 20 mobile applications, as they account for more than 90% of the total bandwidth, as measured from the logs of our dataset. The 20 mobile applications have 11 specific types of services, which are P2P communication; map service; e-mail; video streaming; music streaming; social networking; news; search engine; shopping; advertising; sports media. Of these services, search engines accounted for over 70% of the total number of connections, video and music streaming for 9%, mail for 6%, and social networking accounted for 8% of the connections.

Before the classification, we first visually explore the data to detect its typical characteristics. In Figure 2, we present what we consider as 4 typical daily behaviors in the dataset. The day-oriented pattern displays a high spike of network usage at lunch time (11 a.m to 12 p.m) while the usage drastically decreases afterwards. In our dataset, the majority of these patterns were found in map and location search applications, which may be explained by the fact that people usually use these applications before considering a trip. The night-oriented pattern shows only a slight increase in the usage during lunch hours, while high rate of usage at night time. Applications related to music streaming and sport information often display such a pattern, their usage being concentrated after working hours. The 'camelback' pattern is the most frequent for most of the applications, the usage being concentrated during break hours. The balanced usage pattern is characterized by a regular usage of the mobile application even during working hours. It is mostly observable for high usage applications such as web-search, social networking, and video streaming. Note that two or more different patterns can occur for the same mobile application at different dates, which depends upon a variety of factors such as day of the week, existence of particular events, etc..

4. Data clustering and prediction

In this section, we explain our clustering method of mobile application usage logs, which consists of three phases:

1. Extracting daily profiles of each mobile application usage to represent them in time-series plots.

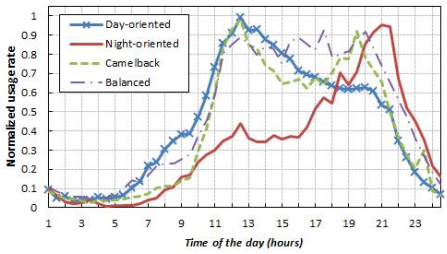


Figure 2: Temporal patterns of users in the dataset

2. Utilization of homogeneous (q-quantiles) and heterogeneous (similarity comparison) clustering techniques to sort and compare daily profiles.

3. Prediction of daily profiles according to their usage pattern.

The functional diagram of our analysis methods are depicted in Fig. 3, which also shows how we explain them in the following subsections. The input data is the raw data list of all URL connection logs per application, while the output is one or more time-series profiles where each resembles a specific class. Using the classes generated from the three phases, we also propose a method to predict future usage for a specific application. It is worth mentioning that the proposed methodology is such that we do not require any training data on a per-user fashion. Instead, we work on data aggregated at a local scale and we build classes of behaviors, which are supposed to be robust through time. Therefore, our analysis is only marginally dependent of the arrival and departures of users in the network.

4.1. Profiling daily usage of mobile applications

In the first phase, we extract the daily usage of each mobile application from the raw data. Let us precise here the definitions that we use in the following.

- A 'profile' denotes any collection of the number of connections to a mobile application as a function of time. In our work, we used 'daily profiles' , that is a collection of the number of connections to a mobile application on a specific day.

- 'Clustering' is the process of repeatedly aggregating daily profiles until obtaining few representative clusters, which we define as 'classes' in this context.

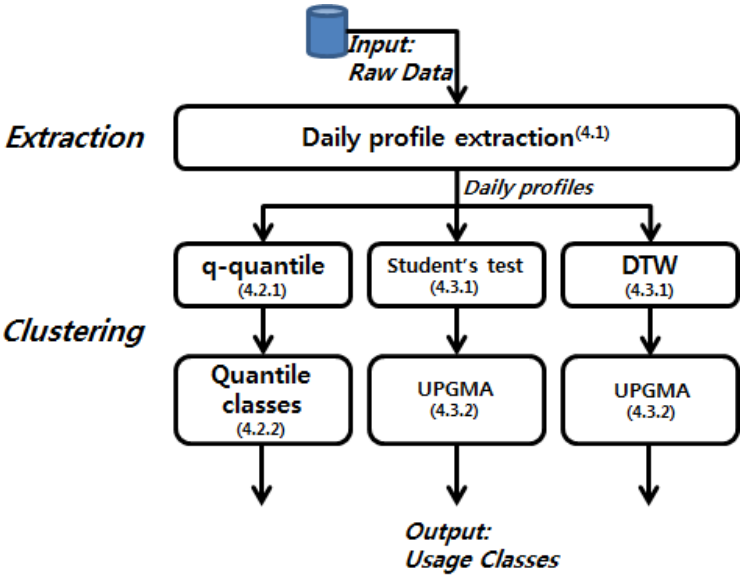


Figure 3: Functional diagram of the proposed data clustering

- A 'class' is a set of profiles resulting from a clustering method. It is represented by a profile which is the mean of the 'daily profiles' belonging to the class.

For each day, the number of usages per application is summed up in every 30-min intervals, and then represented in a time-series graph as shown in Fig. 4. Since our analysis set includes four months from March to June, we generate 122 profiles for every mobile application. Fig. 4 shows a visual example of web-search application usage to how each daily profile can be generated and depicted as a time-series graph. Each usage profile represents a specific date, with the x-axis denoting hour of the day and the y-axis representing number of connections per interval. As shown in the figure, all graphs have distinct behavior according to the number of connections.

4.2. Homogeneous clustering using q-quantiles

After the creation of daily profiles, for each application, levels of similarity for the 122 daily profiles (i.e., how similar they are to each other) are measured using various similarity comparison techniques. Firstly, to distinguish the volume usage between profiles, we utilize quantile-based clustering [23].

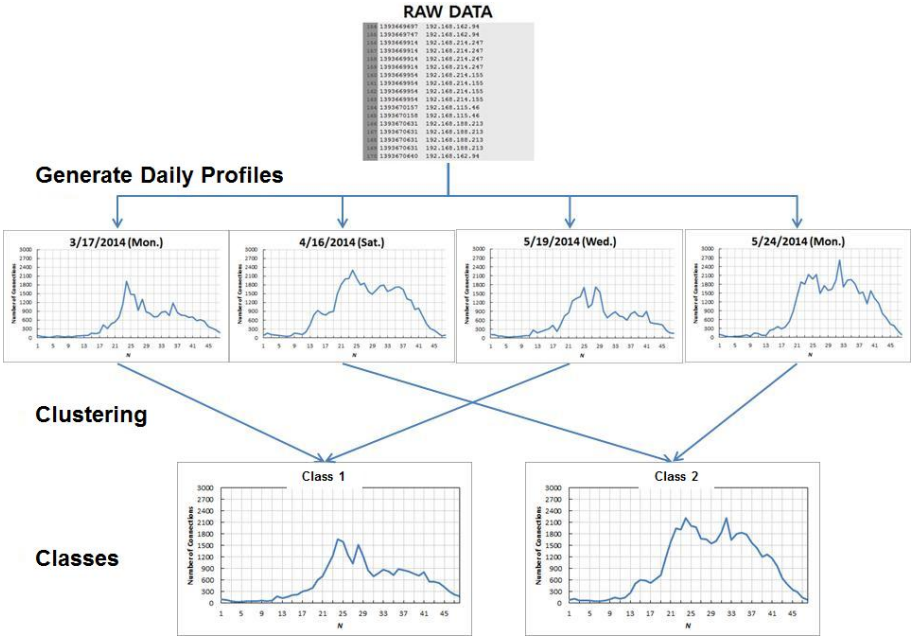


Figure 4: Visual example of generating daily profiles and classes

q-quantile method is a lightweight approach of clustering, as it simply demands to rank profiles according to their total usage volume. Compared to traditional similarity-based methods, it guarantees a lower computational complexity, i.e. $O(M \cdot logM)$ in case quicksort is used, where M is the number of daily profiles that need to be sorted. One characteristic of q-quantiles is that the number of profiles are evenly distributed into each quantile, making the size of each quantile homogeneous.

4.2.1. Quantile-based clustering

To derive specific classes from multiple profiles, we need to be able to insert profiles that are resembling each other into a same class while differentiating those that are clearly distinct. To identify the similarity and the difference between any two given profiles within a same mobile application, we utilize quantile-based clustering according to the usage volume of each daily profile. Let $x_t^k(a)$ be the collection of all t-th interval from k-th profile in application a, $\tilde{x}_t(a)$ be the median of x_{ta}^k series over k, and P_a be the

resulting "median profile" with $\tilde{x}_t(a)$ as temporal components, then:

$$P_a = \{\tilde{x}_1(a), \tilde{x}_2(a), ..., \tilde{x}_N(a)\} \tag{1}$$

We sort each daily profile k according to the overall difference with respect to P_a. Let $x_t^k(a)$ be the volume of t-th interval in profile k. The difference of volume that we define as "volume gap", for every profile k in an application a is $g_k(a)$ computed as:

$$g_k(a) = \sum_{t=1}^{N} (x_t^k(a) - \tilde{x}_t(a)) \tag{2}$$

where t denotes the t-th interval of the profile k and median profile P_a. Thus, $g_k(a)$ becomes a collection of volume gap values for all profiles. Using this method, we can sort all the profiles in the respective order and apply different q for selecting the number of quantiles. In our work, we use $q = 2, 4, 8$, which are median, quartile, and octile, respectively. For example in the case of $q = 4$ (we can calculate the cutpoints for quartile 1, 2 (median) and 3), allows us to retrieve 4 distinct classes. We note that clustering using q-quantiles guarantees equal number of profiles per class.

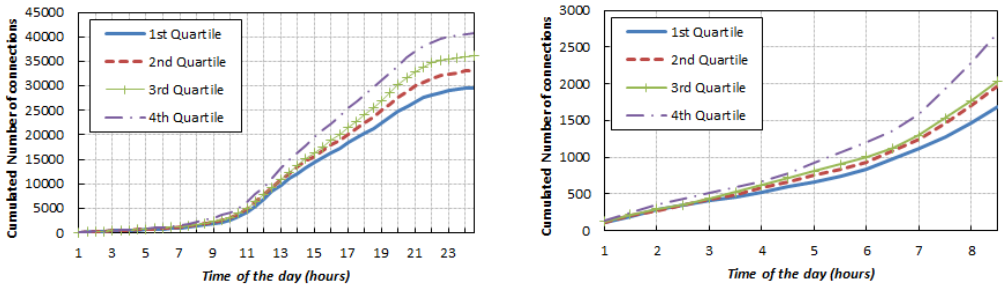
4.2.2. Class generation

After configuring the range of each quantile, we can derive the representative profile of each quantile, which will act as one of the classes for the application. Let c'_t be the collection of t-th interval in quantile c. The representative profile of quantile c, P^c, is the mean value of all daily profiles that belong to the respective quantile, as shown below:

$$P^c = \{\bar{c}'_1, \bar{c}'_2, ..., \bar{c}'_t\}, \tag{3}$$

For any given application, the number of classes will result in the same number of q used to configure the number of quantiles. For example, Fig. 5 shows the result of 4-quantile (quartile) clustering of a web-search application, with each class represented in the form of a cumulative curve.

As seen in Fig. 5, the pattern of each class are easily distinguishable because the gap in the number of connections are distinctively clear. Basic intuition is that usage of any application tend to have dramatic increases when $N > 22$ (after 11:00 am) and gradually decreases when $N > 40$ (after 10:00 pm). However, it is evident that the rate of the increase for each day



(a) N = 24 hours

(b) N= 8 hours

Figure 5: Creation of classes using 4-quantile (quartile)

can be completely different. For example, in the case of Fig. 5 (a) the total usage in the fourth quantile is more than 30% higher than the usage detected by the first quantile for the web-search application.

It is important to note in Fig. 5 (b) that even though we clustered each daily profile according to their total usage volume, when we zoom into the first 8 h, we can observe that there are differences that can be distinguished between the classes. This leads us to believe that we can utilize a specific time t to estimate and predict what the usage of the application will be in the next $N - t$ hours.

4.3. Heterogeneous clustering

Even though q-quantiles can create classes of equal size, it does not efficiently account for specific applications that may have more dynamic behavior. This feature especially becomes a drawback when applied to network contexts with a heterogeneous distribution of application usages. Therefore, we resort to more sophisticated techniques to create heterogeneous classes. These methods are based on the evaluation of the similarity between two profiles using tests such as Student's test [26] and dynamic time warping (DTW) [27].

4.3.1. Student's test and DTW

Student's test can be used as a basic similarity test between two sets of data, based on the differences of their means. In our work, we perform the paired difference test using Student's test to compare the mean difference of each profile to every other profile and cluster them.

On the other hand, DTW is an algorithm for measuring the similarity of two time-series. DTW is one of the most favored choices for time-series comparison in the literature, as shown in many existing works [24] [25]. It is a method already utilized in various applications such as speech recognition, automated signature recognition and shape matching. In our work, we calculate the DTW between two daily profiles and then sum the results, which return a single integer value that represents the overall distance between the two profiles.

4.3.2. Class generation using clustering

When Student's test and DTW is used, the basic assumption is that profiles with low distance scores should be clustered together, so the strategy is to find the lowest distance pairs among all profiles and cluster them in this order. To achieve this goal, we utilize the Unweighted Pair Group Method with Arithmetic Mean (UPGMA) [4] [22]. Using UPGMA, it is possible to cluster all the profiles until only one is remaining. Authors of [4] have already utilized UPGMA for application in mobile phone call analytics, and shown that it can be utilized in differentiating distinct classes of daily profiles. The application of UPGMA in our work runs in three steps: 1) finding the lowest distance pair, 2) pairwise aggregation, and 3) recalculation of distance.

1. *Finding the lowest distance pair:* Firstly, all profiles from the same mobile application are compared with each other using the selected similarity comparison technique. The two profiles that return the lowest distance score (highest similarity) are selected for aggregation.

2. *Pairwise aggregation:* The aggregation procedure follows the unweighted pair grouping of two profiles. If both profiles have never been aggregated, the average of the two time-series profiles are calculated to produce a new aggregated profile. If either or both profiles are the results of previous aggregations, then the mean of the two profiles are weighted according to the number of aggregations that each profile has experienced.

3. *Recalculation of distance:* When a new profile is created from the aggregation process, its distance is recalculated with all other remaining profiles. When the calculation is finished, the lowest distance pair is selected again from either the new profile or the calculation of existing profiles. Pairwise aggregation and recalculation of distance is repeated

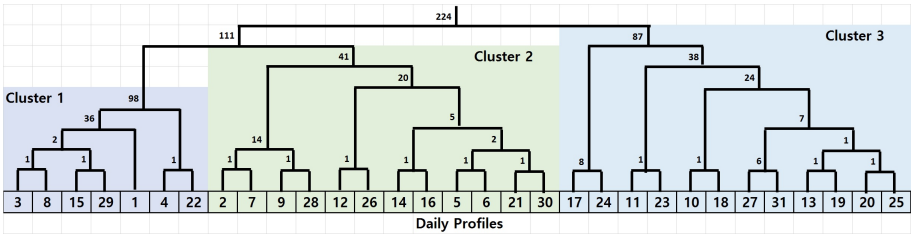


Figure 6: An example of UPGMA clustering of profiles

until there is only one profile remaining. We discuss in the following subsection how to select relevant classes from the UPGMA tree.

An example of a result of three previous steps can be observed in Fig. 6. As shown in Fig. 6, the heterogeneous clustering using Student's test and DTW yields classes that are uneven in size, which can better reflect the distribution of the profiles compared to q-quantile based clustering, which evenly balances the number of profiles per class. However, note that DTW induces a computational cost in $O(N^2)$, where N is the number of measurement points within a daily profile, which can be high in practical contexts (e.g., network controllers). Notice that even with a cheaper comparison method such as Student's test with cost of $O(N)$, the overall time complexity may be critical, as UPGMA depends on the number of daily profiles as well. As any profile must be compared to all others, the clustering method is in $O(M^3)$ where M is the number of daily profiles. As a result, this makes the overall computational cost of Student's test in $O(NM^3)$, and DTW in $O(N^2M^3)$.

4.3.3. Threshold configuration

Like the q-quantile clustering, the UPGMA clustering tree can be statically cut to configure a specific number of clusters. However, as different mobile applications have different usage behavior, it is important for each mobile application to maintain a certain number of classes that is ideal for each. To do this, we propose a threshold configuration method to configure a number of classes for each mobile application. Each pairwise aggregation result returns the lowest distance score that was recorded during one round of calculation. Therefore, after all daily profiles have been clustered, we can obtain a list of distance scores. It is natural to assume that the distance scores acquired at the latter stages of the clustering tend to be higher than the values acquired at the former stages. Also, as Fig. 6 suggests, the last

stages of the clustering occur with distant clusters, which will guarantee much higher distance scores. In fact, our studies show that the list of the distance scores show an exponential increase.

Using this observation, we devise a method for configuring a threshold that allows to prevent aggregation of profiles/clusters that causes a large increase in the distance score. In other words, this helps us to generate multiple number of classes that are distinct from each other. To calculate this, we use the following test of as:

$$\left|d_s - \bar{d}\right| > \sigma \tag{4}$$

where s is the number of all distances generated from the clustering process, d_s is each comparison score, and $\bar{d}$ is the mean of all comparison scores. Therefore, if the test is larger than the standard deviation σ, then the two profiles are too distant from each other to be aggregated and therefore should belong to different classes. By using this method, we can calculate a threshold value that is relative to each mobile application, without any static parameters.

4.4. Prediction using generated classes

Results from the clustering in Fig. 5(b) show that the gaps between classes are wider with respect to time, making the distinction possible with relatively small values of t. This is important because if we can use a small value of t to predict the rest of the day, we can forecast how the rate of the data usage increase will be when it is at its peak, allowing the service provider to take proactive measures to account for it. To do this, we propose a method to predict the class of behavior we should observe on a specific day according to the trend at time t.

When we have the application usage sample from $N = 1$ to $N = t$, this sample can be compared to each of the classes. To do this, we utilize Euclidean distance measure between the cumulative curve of the sample and the classes. If c_t is the t-th entry in class c and s_t is the t-th entry in test sample s, the distance is simply:

$$d(c, s) = \sqrt{\sum_{t=1}^{N} (c_t - s_t)^2} \tag{5}$$

For each class, the distance is measured and the class with the lowest

distance to the test sample becomes the predicted class. Here, we note that for a small value of t, the usage difference between the classes is low as it is early morning and the human activity is also low, making the prediction less reliable. On the other hand, a high value of t allows for a more accurate prediction of the future application usage. However, if the value of t is too high (i.e. exceeding the point of time when the data usage is already peaked), then the objective of forecasting itself becomes invalid because the critical point for adapting to usage increase is already past. One of the main objectives of prediction using an early time of the day is to accurately account for the sudden increase of data during the hours with high activity. Therefore, it is important to find the appropriate value of t that maximizes the accuracy of prediction while meeting the user demands of the application.

When the classes are generated from using Student's test or DTW, then the corresponding similarity comparison technique is used to compare the distance between the test sample and each of the classes from $t = 1$ to $t = N$. Naturally, the correct class that a test sample should belong to is the one that has the lowest distance when compared with $t = N$.

5. Performance analysis

In this section, we conduct extensive evaluation of our proposed clustering and prediction methods. In the first part, we analyze the results from our clustering process. Second part analyzes the accuracy of the classes that we generate by applying k-fold validation technique.

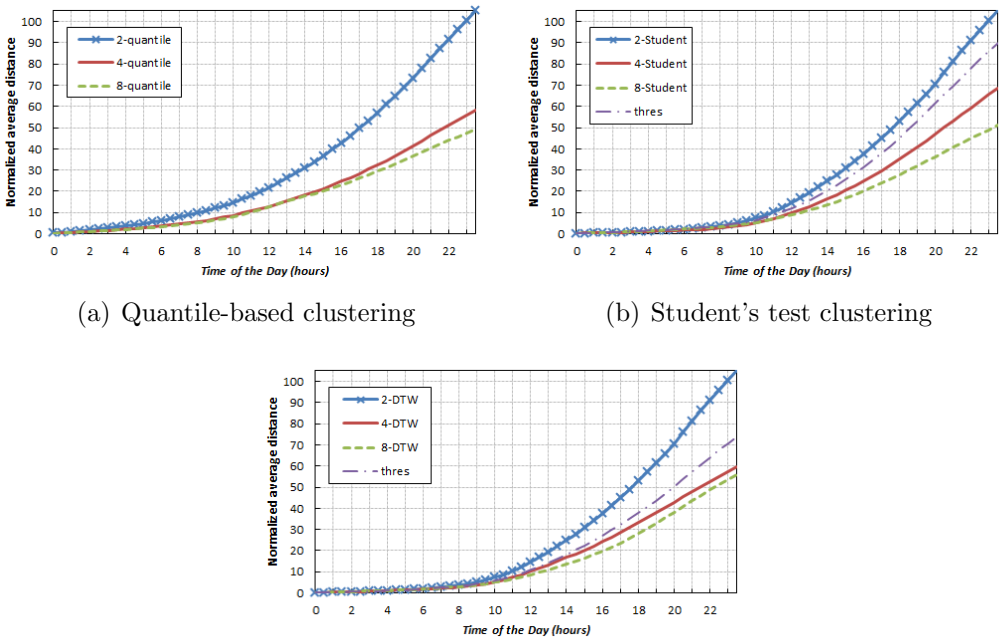
5.1. Clustering results

Firstly, we observe the results of clustering using the quantile-based partitioning, Student's test, and DTW similarity comparison.

5.1.1. Average distance between classes

For the first test we compare the distance between classes when the value of q varies. For all applications, we calculate the Euclidean distance for each t between the classes that are adjacent (lowest in distance) to each other, and then normalize the distance for each application because the usage rate of each application is different. The results are shown in Fig. 7.

Intuitively, it is easier to distinguish between different classes when the number of generated classes is low. This can effectively reduce the value of t needed to predict a specific test sample to belong to a certain class. As



(a) Quantile-based clustering

(b) Student's test clustering

(c) DTW clustering

Figure 7: Normalized average distance between classes

seen in Fig. 7(a), the average distance between the adjacent class gradually decreases as the value of q becomes higher. This makes excessively high value of q undesirable.

However, choosing the lowest value of q does not guarantee the best possible results, because low values of q clusters daily profiles which may be quite different from each other. This in turn possibly increases the difference between the average behavior in the predicted class and the actual usage of the test sample. Therefore, we need further observation of the effect of q on the overall prediction accuracy. In the cases of Student's test and DTW, as shown in 7(b) and 7(c), the distance between classes when threshold configuration is used is between $q = 2$ and $q = 4$, which allows the classes to be more distinguishable from each other while having classes which gather relatively homogeneous behavior.

5.2. Prediction accuracy

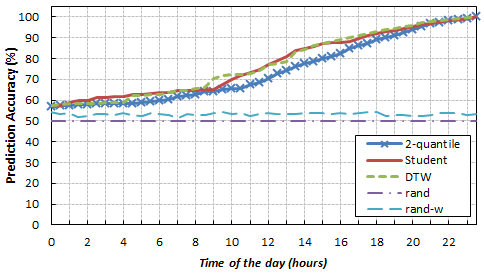
We present an analysis of the accuracy of our proposed mobile application clustering methods. For the first analysis, we use the profiles from March to June, 2014. Therefore, there are 122 profiles for each mobile application. Each profile is considered as a test sample, using the 121 remaining profiles to define the classes. This is to ensure that each one of the profiles can be predicted without it being previously referenced in the clustering process.

We define an accurate prediction as follows. A test sample profile that was not used for the clustering needs to be classified into one of the classes using a limited time t. When quantile-based partitioning is used, Euclidean distance measure is used to compare the corresponding daily profile with all classes. If the closest class is the class that it should belong to using the clustering process, then the prediction is accurate. In other words, assessing the 'prediction accuracy' gives the network controller an intuition of how many hours it has to observe the traffic in order to evaluate the traffic usage during the rest of the day. In the cases of Student's test and DTW, prediction is correct when the selected classes using $t < N$ is the same as the prediction when $t = N$. We compute the accuracy with all profiles and analyze the percentage of accurate predictions from the total number of predictions.

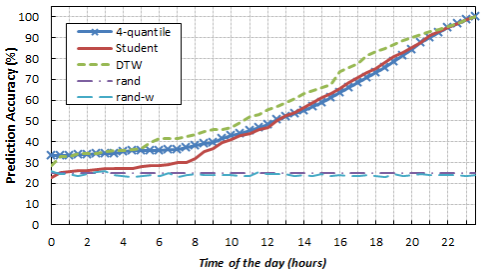
5.2.1. Accuracy sensibility analysis

Fig. 8 shows the performance of our proposed prediction method when the value of t varies from 1 to N for all applications. That is, the prediction accuracy of all 20 mobile applications are separately calculated, then combined and averaged in Fig. 8. Here, N is defined as 48, which means that every graduation on the x-axis in N represents a 30-min interval. The values of q are 2,4, and 8, which denotes the q-quantiles that will be used to create q classes for all clustering methods. $rand$ denotes the random classification for homogeneous clustering, therefore each class has an even chance of being randomly selected. $rand-w$ is the weighted random classification for heterogeneous clustering, where each class has a chance to be selected depending on the number of profiles used to produce the class.

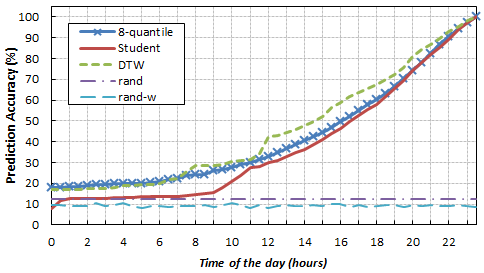
As mentioned from above, a high enough value of t is required to accurately predict the appropriate class for a test sample, but a lower t value is desirable to meet real-time user demands. As seen in Fig. 8(a) when $q = 2$, the prediction accuracy for $t \geq 24$ (12:00 pm) is over 70% for all methods, while reaching over 90% for $t \geq 36$ (18:00). When the value of q increases, the prediction accuracy also falls as shown in Fig. 8(b) and Fig. 8(c), mainly



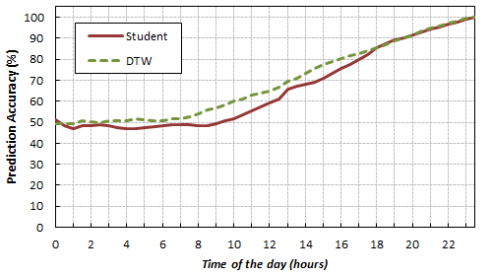
(a) $q = 2$



(b) $q = 4$



(c) $q = 8$



(d) Student's test and DTW with threshold configuration

Figure 8: Evaluation of prediction accuracy

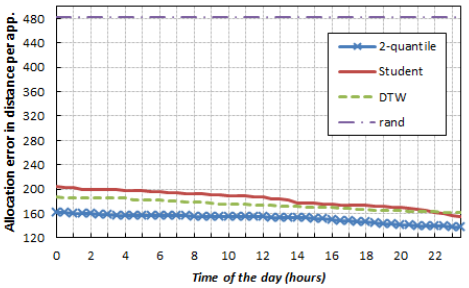
due to the fact that there are more classes with less distance between each other which causes more misclassifications. In general, the performance of the heterogeneous clustering techniques is comparable to the performance of homogeneous clustering, with Student's test performing about 10% lower in the worst case than DTW. In overall, the performance of all three methods greatly outperform random classification methods. We note that when applying this method for resource allocation on a WLAN network, the service provider should compute an appropriate t for each type of application, depending on the demand of each application, to maximize QoS. Fig. 8(d) shows the performance of Student's test and DTW methods when the threshold configuration in section 4.3.3. is used. Even though the number of classes can vary per application, the threshold configuration manages to set an appropriate number of classes, enough to guarantee prediction accuracy as high as when $q = 2$.

From the results in Fig. 8, it seems that a lower value of q would be

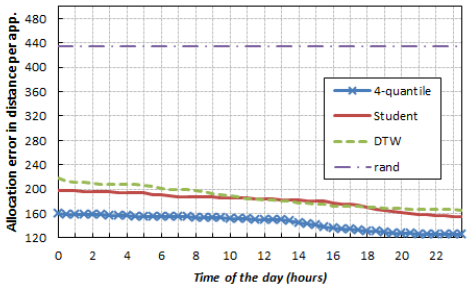
more efficient than higher values of q because it can increase the prediction accuracy. However, as stated above, the larger distances between classes can result in considerable distance between the correctly-guessed class and the test sample. To observe this effect, we calculate the distance, which we define as "allocation error". In Fig. 9, the y-axis, which we denote as allocation error per mobile application, represents the absolute difference from the selected class's number of connections to its actual number of connections in the profile considered. We assume that an access network provider decides on allocating bandwidth depending on the result of the prediction. If an incorrect prediction is made, the chance of over-allocation (allocating more bandwidth than the actual usage) or under-allocation (allocating less than the actual usage) of bandwidth for the specific application would also increase. Also, even if the prediction made is correct, the distance between the prediction and the actual data usage can still be large. Therefore, a scheme that allows lower over/under bandwidth allocation can be considered more accurate than other methods. Here in our work, we first observe the effect of q on the bandwidth over/under allocation (as defined by allocation error), and also compare our proposed prediction methods with average prediction. Average prediction calculates the average of all profiles for each N and creates the corresponding profile. The results can be observed in Fig. 9, which shows the average bandwidth allocation error.

Note that we omit the results of $rand-w$ because in all cases it performed worse than the unweighted random case. As seen from Fig. 8 and Fig. 9, high prediction accuracy does not necessarily translate into efficient allocation of bandwidth. For example, $q = 2$ which had the highest prediction accuracy in Fig. 8 actually has higher distance between the test sample and the predicted class when the value of t becomes higher. On the other hand, higher values of q provide lower distance, which means that even though the actual prediction is incorrect, in terms of guaranteeing the correct bandwidth, it performed better as the value of t increases. This is evident for q-quantiles, where in the case of $t = 1$ allocation error of $q = 2$ is 160 and error of $q = 8$ is about 180, while in case of $t = 48$ allocation error of $q = 2$ is 140 while error of $q = 8$ is 120.

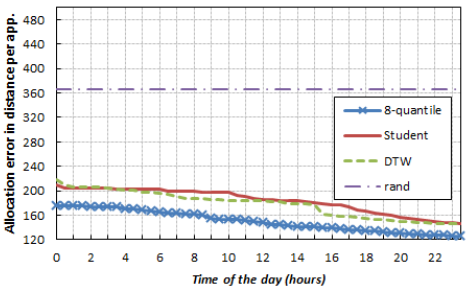
All proposed methods of prediction significantly outperform the random classification method. However, it is important to note that allocation error of both heterogeneous clustering methods are considerably higher than the homogeneous counterpart in all cases of q. We believe this phenomenon results from two reasons. The first reason is that fixed value of q is not suitable



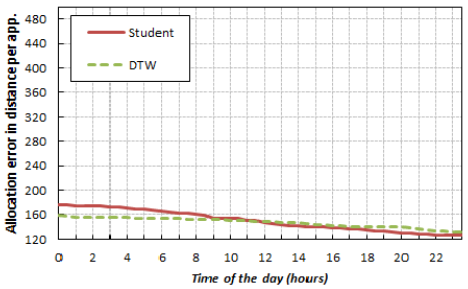
(a) $q = 2$



(b) $q = 4$



(c) $q = 8$



(d) Student's test and DTW with threshold configuration

Figure 9: Evaluation of allocation error

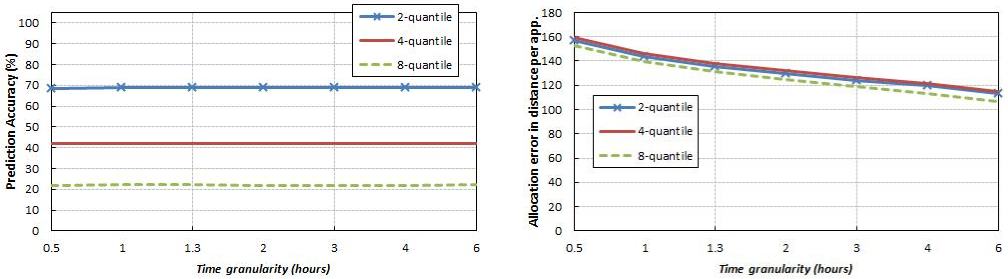
for heterogeneous clustering which forces slicing a class which should be one, or sometimes aggregating classes that should be separated. Another reason is due to outliers, where we found that without any outlier control, Student's test and DTW can sometimes generate small classes that are actually outliers. This causes higher error in the overall prediction.

The first reason can be easily solved using the proposed threshold configuration. As shown in Fig. 9(d), the performance of Student's test and DTW can be increased through threshold configuration. Especially for DTW, compared to $q = 4$, the allocation error is about 10% better when $t = 1$, and 20% better when $t = 48$. The reason for this increase is because the threshold configuration allows classes of different sizes, where sizes are optimal in the sense that profiles of a class are supposed to be as similar as possible.

Note that achieving high performance according to both quality predictors (accuracy, allocation error) is important. As seen in Fig. 9, higher prediction accuracy is required to maintain low distance between the test sample and the predicted class, as low q for clustering methods guarantees lowest distance when t is low. However, as t becomes higher and prediction accuracy for $q = 8$ also increases, its distance becomes the lowest as it allows more refined prediction using more classes. Therefore, finding the appropriate tradeoff by tuning q becomes an important issue which deserves further study in future work.

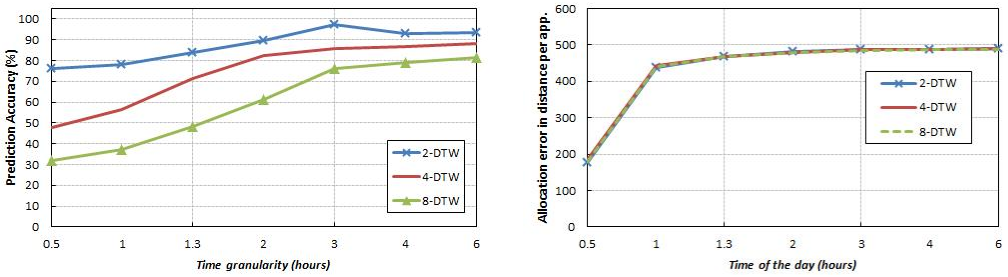
5.2.2. Time granularity analysis

Fig. 10 shows the differences in performance regarding different time granularities. Here the horizontal axis represents the duration of a sample. For each test case, the accuracy and the allocation error is calculated at 12:00 pm. As different time granularities does not affect the q-quantiles clustering, which only depends on the overall volume usage, the prediction accuracy does not depend on time granularity, as can be seen in Fig. 10(a). However, Fig. 10(b) shows that the allocation error actually decreases with larger time slots. The reason for this is that the behaviors are smoothed with larger time granularities, resulting in smaller over/under allocations. The allocation error is lower with a higher time granularity for q-quantile, however, increasing the granularity reduce the number of time slots for predictions (e.g. 12-hour granularity only allows the controller to make one decision per day). On the other hand, Fig. 10(c) shows that for DTW the accuracy can be significantly increased when larger time slots are used. The classes generated seem to be more distinct. However, as shown in Fig. 10(d), the performance in terms of



(a) q-quantile prediction accuracy

(b) q-quantile allocation error



(c) DTW prediction accuracy

(d) DTW allocation error

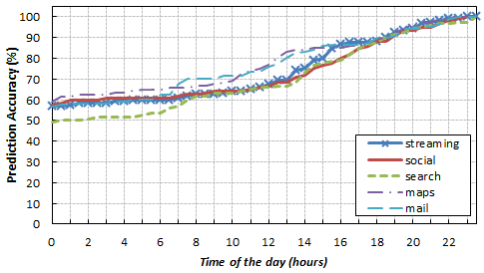
Figure 10: Evaluation of time granularity

allocation error is degraded. Profiles being clustered with a rougher grain, a classification error translates in a larger allocation error. Therefore, in case of DTW, using a finer granularity is the more efficient choice for our dataset.

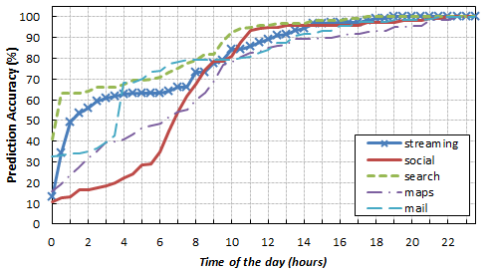
5.2.3. Application-level prediction accuracy

Even though the performance of the heterogeneous clustering methods can be improved using the proposed threshold configuration method, outliers in profiles still affect the quality of the prediction process. To observe this, we analyze the application-level prediction accuracy and the normalized allocation error, which is shown in Fig. 11. Here the normalized allocation error refers to the performance scale of Student's test when the performance of q-quantiles is normalized to 1. For simplicity, we show the results of five different applications, using $q = 2$ for quantile-based clustering and Student's test.

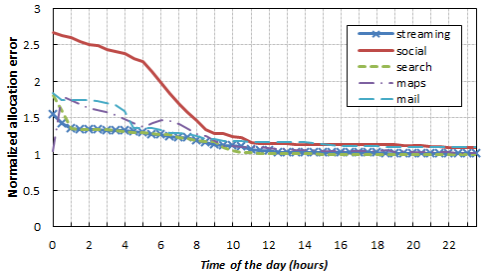
As shown in the figure, the prediction method depending on value t can



(a) Accuracy of 5 applications using 2-quantile



(b) Accuracy of 5 applications using Student's test



(c) Normalized allocation error of Student's test

Figure 11: Prediction accuracy and allocation error of types of services

dynamically affect the prediction accuracy of each type of service, thus showing again why different applications require different t for more accurate predictions. We can also observe that quantile-based partitioning guarantees a relatively stable performance for all applications, while Student's test has high fluctuation depending on it. Fig. 11(b) shows that the accuracy of certain applications, such as social and maps mobile applications, have relatively lower performance. This is also reflected in Fig. 11(c) where the allocation error for predicting social network application is more than 2.5 times higher.

The main reason for this difference even though $q = 2$ for both methods is because Student's test is more heavily affected by outlying behavior than the homogeneous method of q-quantile clustering. The heterogeneity of the classes using UPGMA makes some classes with low number of profiles becomes more affected from an outlier. In some cases, an outlier may be one of the last profiles to be clustered in the UPGMA, possibly creating a outlier class. Our studies showed that in Fig. 11(b), one of the only two classes was in fact an outlier which had similar appearance to the other classes in the early hours of the day. Therefore, many predictions in the early hours were wrong and the performance was severely affected.

6. Challenges in detecting and qualifying outliers

As noted before, one of the reasons that the proposed clustering and prediction schemes may malfunction is due to outlying behaviors in a daily profile. An outlier in a daily profile corresponds to an unexpected behavior that are different from former observations. To observe outliers in our current dataset, we modify the homogeneous and heterogeneous clustering methods according to their characteristics. Note here that our work focuses on how to identify outliers, then we exclude them from the clustering process in order to evaluate their actual impact on potential allocation error.

6.1. Outlier detection using q-quantiles

We utilize quantile cutpoints and the interquartile range (IQR) to distinguish outliers. However, instead of defining outlier events just from the total volume of each profile, we make the comparison on a 30-min interval basis. The reason why we utilize such approach is to account for sudden fluctuations of volume usage within each daily profile, which occurs frequently but cannot be predicted in the total usage volume of each day. For simplicity, we detail the outlier detection in 4-quantile (quartile) division, although it can

be used in the same way for other values of q. Using each $\tilde{x_t}$ in P_a shown in (1), an outlying 30-min interval is measured for each profile k. Let $Q_{3,t}$ be the third quartile and $Q_{1,t}$ be the first quartile for the interval t, then IQR of interval t, IQR_t, is defined as

$$IQR_t = Q_{3,t} - Q_{1,t} \tag{6}$$

Using the IQR_t, we configure an upper bound U_t and the lower bound L_t to exploit an outlying event,

$$\begin{aligned} U_t &= Q_{3,t} + 1.5(IQR_t) \\ L_t &= Q_{1,t} - 1.5(IQR_t) \end{aligned} \tag{7}$$

Thus, we gain the lower and upper bounds for each interval of t for all profiles. Here, for each interval t for profile k, if the number of connections at t is higher than U_t or lower than L_t, then that specific interval t is considered as an outlying event and the number of connections corresponding to that t is recorded. For an upper bound outlier, the number of connections is recorded in a positive integer, while for a lower bound outlier, it is recorded in a negative integer. Therefore, each profile k will end up with a outlying number of connections O_k. From O_k and the median, we can recalculate the first and third quartiles and acquire the $IQR\prime$. In the same way, upper bound and lower bound U and L can be acquired. This enables us to detect the outlier k if the O_k is lower than L or higher than U. Note that the weight given to IQR_t is an empirical value which can be adjusted to control the number of outliers.

We apply the outlier detection method on the top 20 used applications in our dataset to discover specific events. Some outliers could be explained by *ad hoc* interpretations, especially for sports classes where outlying dates in sports application coincided with the European champions league final (May 24, 2014), and world cup games held in June. Also, social networking applications had outliers that coincided with city scale elections in France, as well as the European elections. We can therefore suggest that these events triggered an unusually high amount of logs for the related applications. Even though this approach could detect some intuitive events, a majority of the outliers could not be easily related to a specific date or event. A major reason for this is that our dataset is based on a WLAN scale volume usage, which

may not always reflect nation-scale events as crowds belonging to different WLAN scale areas may behave differently depending on their tendencies. Furthermore, different applications can be differently affected by the type of event, meaning that a specific event may affect the usage of an application but may also not affect another majority of them. Finally, many outliers cannot be associated to a precise event, because this event may be unknown to us or simply does not exist, as outlying behaviors may stem from mere statistical fluctuations. Therefore, to isolate possible outlying events in a more reactive manner, we utilize the IQR range from the classes that are generated in the clustering process.

For example in the case of $q = 4$, the IQR value can act as a threshold to single out profiles that are too far away in volume usage from the first and last quartile cutpoints. In the same way, IQR value can be redefined by the length of the volume of first and last classes generated from the quantile-based clustering. When t denotes the number of 30-min intervals, the upper bound and the lower bound can be defined as:

$$\begin{aligned} U' &= C_3[t] + 1.5(C_3[t] - C_1[t]) \\ L' &= C_1[t] - 1.5(C_3[t] - C_1[t]) \end{aligned} \tag{8}$$

where $C_1[t]$ and $C_3[t]$ are the values of classes one and three at instant t. For the test profile, if the value of the t-th entry is higher than U' or lower than L', then it is considered as an outlier. We implement this method in our proposed clustering technique using quantiles and compare the results with Fig. 11. When the outlier detection is used, the detected outliers are isolated and not allocated to any class, reducing the error in over/under allocation of bandwidth. The results are shown in Fig. 12.

As shown in the figure, the allocation error is significantly reduced. This is because outliers are one of the main reasons for incorrect allocation of bandwidth. By eliminating outliers before making erroneous predictions from one of the classes, bandwidth error can be reduced more than roughly 10% for both $q = 2$ and $q = 10$.

6.2. Outlier detection using UPGMA

For Student's test and DTW, we can utilize UPGMA to exploit outliers in the clustering phase. The same method as the one defined in (4) can be used to test the variance of all 122 original profiles when they are clustered.

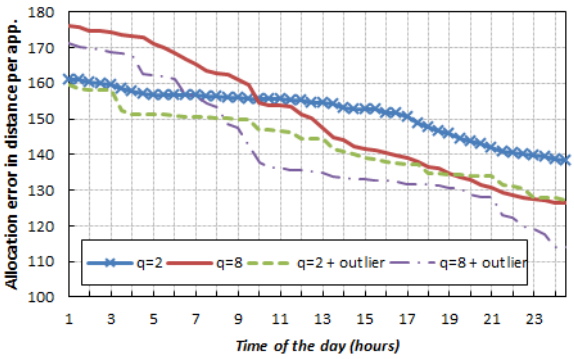
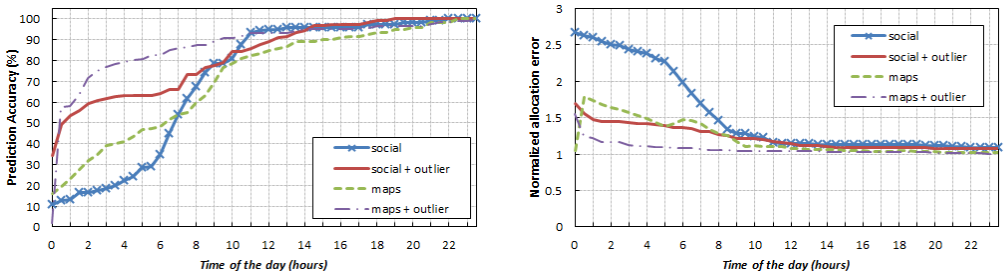


Figure 12: Allocation error comparison using outlier detection

To do this, instead of listing the list of all comparison scores as done in the threshold configuration, we only list the comparison scores of the original profiles that are not yet clustered. Then, using (4), we calculate the standard deviation σ and the mean difference M_s of the original profiles. If M_s of an original profile is higher than σ, then it is considered as an outlier. This can effectively distinguish outlying behavior because outlying profiles tend to be clustered in the later stages of UPGMA, as they have higher distances to all other profiles.

We apply our outlier detection using UPGMA and observe the improvement compared to the results in Fig. 11. The improvements in prediction accuracy and allocation error results for social and maps mobile applications are shown in Fig. 13. From Fig. 13, we can observe that eliminating outliers during the clustering process can greatly improve the overall prediction accuracy and allocation error. Especially in the case of maps application, eliminating outliers result in more than 40% increase in prediction accuracy and 40% decrease in the allocation error when $t = 4$. This is because one of the classes that were selected through UPGMA was an outlier, and the accuracy of the prediction benefits from eliminating it.

We note that in this research, we propose some simple methods of outlier detection. Even though we manage to acquire some promising preliminary results, we plan to further enhance our methodology. Firstly, we believe that we can discover more outlying behaviors on a per-application level basis by examining applications of the same type. Also, as noted above, the detected outliers are currently just isolated in the clustering process and not treated to enhance the allocation prediction. How to cluster and predict the actual



(a) Accuracy of Student's test with outlier detection

(b) Allocation error of Student's test with outlier detection

Figure 13: Prediction accuracy and allocation error using outlier detection

usage resulting from outliers will also be addressed in future work.

7. Discussion and conclusion

Throughout the paper, we have shown how we can cluster and predict how and when users access mobile applications through their wireless communication means. We discuss our methodology and some of the use cases where our contributions could be exploited.

We note that the methods that we designed to profile, cluster, and predict mobile application usage are lightweight methods that are also frequently utilized in other areas of research. However, we also note that our work is focused on exploiting data consumption behavior depending on the temporal application characteristics as well as their type. We believe that our work can be a notable starting point for other researchers that are interested in the analysis of data consumption behaviors through various types of mobile application usage.

One of the biggest characteristics of the dataset that we utilize is that the logs are generated from a WLAN-scale network, which has a small and limited scale in terms of spatial properties and number of users. This differs from the dataset used in city-scale data from existing work such as [3] and [4], especially in the sense that our dataset could be more heavily affected by human stochasticity and could tend to have higher fluctuations in the usage volume. This leads us to believe that each WLAN can consume different types and volumes of data depending on the context of the WLAN

(restaurant, train station, etc.), and that more outlying behavior and differences in classes may occur more frequently. Therefore, our analysis and proposed scheme can be useful for adaptive planning and management of a WLAN-scale network, especially in software-based network controlling as defined in NFV and SDN-type systems.

If we can predict how much users access what applications at which specific times, it becomes possible to pro-actively adjust a network service to meet the user demands. One prominent use case is the design of a Wi-Fi cloud system, which in fact our data was derived from. For example, a SDN-based Wi-Fi cloud consists of a controller that is capable of controlling the connections between the users and the external web servers, acting as a gateway. If the controller can predict the usage of service users, then it is possible to take proactive measures, such as adaptive port control, proactive content caching, and bandwidth adjustment. In edge computing, the application server or network function server capacity can also be adapted as a function of the load, scaling in or out the number of cores, memory, and etc.

The information regarding the correlation of identical applications can also help network service providers to predict the usage of applications that are not yet analyzed. As there is an extremely large amount of network contents and applications that exists on the Internet, it is impossible for the service provider to possibly analyze all the existing contents. Instead, by using the analysis reports of a well-known application, it is possible to accurately predict the usage, bandwidth consumption, and possible congestion that can occur from the new application, allowing the controller/administrator to make the appropriate decisions in adapting to the new usages. Notice that the traffic pattern can change, for example because of version upgrades or sudden popularity trend changes. In this case, our method should allow to change the classes in order to match the new traffic patterns detected. This problem is closely related to event detection in the system, which we will address in future work.

Mobile data analytics have been a vital part of the computing research community over the past decade. Even though many significant works have been made in exploiting spatiotemporal behavior of mobile users, research regarding data usage and web access is limited. For this, we exploit Wi-Fi usage data to profile and cluster human data/content usage behavior, showing how to predict future data usage for different mobile applications. We show that homogeneous clustering method guarantees simple yet effective classification, while heterogeneous clustering provides more effective level of

prediction accuracy. Our evaluation shows that even with a small window of time sampling, we can predict the peak data usage of various applications, which considerably differs in both number of connections and peak usage time. Further work is needed to more profoundly integrate outlier detection in the prediction phase, and evaluate our proposed method in practical use-cases.

Acknowledgement

This project was partly supported by the ABCD (Grant No: ANR-13-INFR-005) and CODDDE (ANR-13-CORD-0017-01) projects funded by the French National Research Agency, and the EU FP7 MobileCloud (Grant No: 612212) project.

[1] Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, *http://www.cisco.com/c/en/us/solutions/service-provider/visual-networking-index-vni/index.html*

[2] J. G. Andrews, S. Buzzi, W. Choi, S. V. Hanly, A. Lozano, A. Soong, J. C. Zhang, "What Will 5G Be?," Journal of Selected Areas in Communications, 32 (6), pp. 1065-1082, 2014.

[3] M. C. Gonzalez, C. A. Hidalgo, A.-L. Barabasi, "Understanding individual human mobility patterns," Nature, 453, 2008.

[4] D. Naboulsi, R. Stanica, M. Fiore, "Classifying Call Profiles in Large-scale Mobile Traffic Datasets," IEEE Infocom, 2014.

[5] D. Naboulsi, M. Fiore, S. Ribot, R. Stanica, "Large-scale Mobile Traffic Analysis: a Survey," IEEE Communications Surveys and Tutorials, Vol. PP (99), Oct. 2015.

[6] ETSI, "Network Functions Virtualisation - White Paper 3," *https://portal.etsi.org/*

[7] ETSI, "Mobile-Edge Computing - Introductory Technical White Paper," *https://portal.etsi.org/*

[8] Z. Qin, G. Denker, C. Giannelli, P. Bellavista, N. Venkatasubramanian, "A Software Defined Networking Architecture for the Internet-of-Things," IEEE/IFIP NOMS, 2014.

[9] M. Seddiki, M. Shahbaz, S. Donovan, S. Grover, M. Park, N. Feamster, Y. Song, "FlowQoS: QoS for the Rest of Us," ACM HotSDN, 2014.

[10] D. Raumer, L. Schwaighofer, G. Carle, "MonSamp: A Distributed SDN Application for QoS Monitoring," IEEE FedCSIS, 2014.

[11] V. Blondel, A. Decuyper, G. Krings, "A survey of results on mobile phone datasets analysis," EPJ Data Science, Vol. 4 (1), pp. 1-55, 2015.

[12] S. Isaacman, R. Becker, R. Ceres, M. Martonosi, J. Rowland, A. Varshavsky, W. Willinger, "Human mobility modeling at metropolitan scales," ACM MobiSys, 2012.

[13] A. Noulas, S. Scellato, R. Lambiotte, M. Pontil, C. Mascolo, "A tale of many cities: universal patterns in human urban mobility," PLoS One 7, 2012.

[14] G. Ranjan, H. Zang, Z. Zhang, J. Bolot, "Are Call Detail Records Biased for Sampling Human Mobility?," SIGMOBILE Mobile Computing and Communication Review 16 (3), pp. 33–44, 2012.

[15] H. Falaki , R. Mahajan, S. Kandula, D. Lymberopoulos, R. Govindan, D. Estrin, "Diversity in smartphone usage," ACM MobiySys, 2010.

[16] J. Kang, S. Seo, J. Hong, "Usage pattern analysis of smartphones," IEEE/IFIP NOMS, 2011.

[17] B. Xiang, K. Zhu, X. Zhang, Y. Yin, L. Zhang, "Exploring Mobile Data on Smartphones from Collection to Analysis," IEEE ICT, 2014.

[18] Q. Xu, J. Erman, A. Gerber, Z. Mao, J. Pang, S. Venkataraman, "Identifying Diverse Usage Behaviors of Smartphone Apps," ACM IMC, 2011.

[19] T. Li, J. Liu, Z. Lei, Y. Xie, "Characterizing Service Providers Traffic of Mobile Internet Services in Cellular Data Network," IEEE IHMSC, 2013.

[20] E. Rescorla, "HTTP Over TLS," RFC 2818, 2000.

[21] D. Eastlake 3rd, "Transport Layer Security (TLS) Extensions: Extension Definitions," RFC 6066, 2011.

[22] K. Kalpakis, D. Gada, V. Puttagunta, "Distance Measures for Effective Clustering of ARIMA Time-Series," IEEE ICDM, 2001.

[23] R. Hyndman, Y. Fan, "Sample Quantiles in Statistical Packages," American Statistician, Vol. 50 (4), pp. 361–365, Nov. 1996.

[24] B. Fulcher, N. Jones, "Highly Comparative Feature-Based Time-Series Classification," IEEE Transactions on Knowledge and Data Engineering, Vol. 26 (12), pp. 3026-3037, 2014.

[25] T. Fu, "A review on time series data mining," Elsevier Engineering Applications of Artificial Intelligence," Vol. 24, pp. 164-181, 2011.

[26] E. W. Weisstein, Student's t-distribution, . *http://mathworld.wolfram.com/Studentst-Distribution.html* 2009. From MathWorld–A Wolfram Web Resource.

[27] T. Rath, R. Manmatha, "Word image matching using dynamic time warping," IEEE CVPR, 2003.