# Affective Contextual Mobile Recommender System

Chao Wu[1], Jia Jia[1], Wenwu Zhu[1], Xu Chen[2], Bowen Yang[1], Yaoxue Zhang[1,3]
[1]Department of Computer Science and Technology, TNList, Tsinghua University, China
[2]Institute of Computer Science, University of Göttingen, Germany
[3]School of Information Science and Engineering, Central South University, China
cwu12@mails.tsinghua.edu.cn, zyx@csu.edu.cn

## ABSTRACT

Exponential growth of media consumption in online social networks demands effective recommendation to improve the quality of experience especially for on-the-go mobile users. By means of large-scale trace-driven measurements over mobile Twitter traces from users, we reveal the significance of affective features in shaping users' social media behaviors. Existing recommender systems however, rarely support this psychological effect in real-life. To capture this effect, in this paper we propose Kaleido, a real mobile system to achieve an affect-aware learning-based social media recommendation. Specifically, we design a machine learning mechanism to infer the affective feature within media contents. Furthermore, a cluster-based latent bias model is provided for jointly training the affect, behavior and social contexts. Our comprehensive experiments on Android prototype expose a superior prediction accuracy of 82%, with more than 20% accuracy improvement over existing mobile recommender systems. Moreover, by enabling users to offload their machine learning procedures to the deployed edge-cloud testbed, our system achieves speed-up of a factor of 1,000 against the local data training execution on smartphones.

## Keywords

Recommender system; affective computing; mobile application; social networks

## 1. INTRODUCTION

The mass-adoption of mobile social networking services and the wide integration of sharing media content on popular mobile applications have paved the way for quantitative research efforts tackling the relations between content and virality [1]. Very recently, a novel kind of user experience is working its way through the online space, i.e., designed to take the affective pulses appeared in online social network (OSN) usage, such interface supports people to explicitly bridge their emotional states with OSN information subscription [2].

To capture users' attention, many tweets in OSNs nowadays are usually published with affective media content [3].


*MM '16, October 15-19, 2016, Amsterdam, Netherlands*

DOI: http://dx.doi.org/10.1145/2964284.2964327

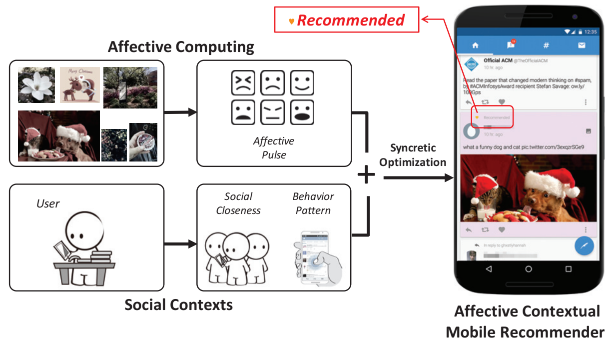



**Figure 1: Conceptual Kaleido diagram.**

In real-life usage, due to the *affective pulse* [4], i.e., the first feeling when accessing an object, triggers the mind 3,000x faster than rational thoughts, many users sensibly make a quick decision whether to subscribe a media tweet by such affective pulse (e.g., happiness, sadness, or disgust) [5]. This actually opens up a new venue to integrate the affective feature for future media recommender system design. Indeed, by means of large-scale data-driven measurements in our early stage work, we reveal that 60% user clicks are motivated by media contents, among which, more than 76% are triggered with explicitly affective pulses (§2.2). Moreover, with further affect-aware measurement in our established system, users are benefited with 82% accuracy to subscribe the right tweets (§5.3). Therefore, it is highly promising to rethink the social media recommender mechanism by jointly considering users' affective pulses as well as traditional features.

Existing recommender systems, however, provide recommendations largely based on users' content preference, using content- [6], demographic- [7], knowledge- [8], or utility- [9] based methods. To boost the prediction accuracy, recent studies [10] and [11] proposed methods which input the user's OSN usage pattern into a linear regression for prediction. To further improve social media propagation or streaming delivery, some researchers try to use both user preference and network context to determine the appropriate presentation method [12]. However, we have yet to see an approach that shifts mobile social media recommendation from the above approaches to a scheme that jointly tackles user's feeling, which plays a critical role in media content consumption in OSNs, behavior pattern (i.e., user preference, content attributes, media formats, time, and network) and social closeness (i.e., social interaction strength).

To fill this void, in this paper we propose a real system *Kaleido*, which jointly utilizes the unique visual features (to capture user's affective pulse), mobile behavior patterns and social friendship closeness, in OSNs for mobile media rec-

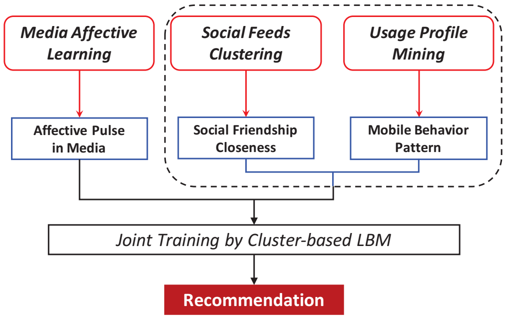



**Figure 2: Framework of Kaleido system.**

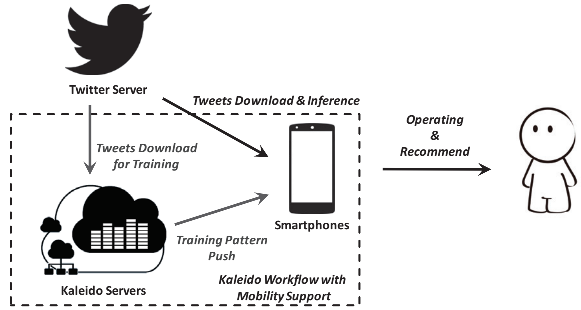



**Figure 3: Logical workflow of Kaleido.**

ommendation. Let us consider the case of Figure 1 where, by integrating Kaleido into a third-party Twitter app, i.e., Twidere [13], the user's social media experience is benefited from her habits along both affective (i.e., happiness) and social closeness (i.e., keep eyes on the sender) directions. To this end, we propose a learning-based scheme to infer affective pulses from media files. On this basis, we combine the affective feature with user behavior pattern and social friendship to further predict users' potential interests in media usage. Moreover, we employ such inference and network environment (e.g., WiFi available or not), to execute the whole prediction.

Specifically, to better understand affective pulses in social media, we employ a learning-based affective computing mechanism and a Flickr image dataset with well-known ground-truths, which has been manual tagged with affective tags by prior work [14], to infer the probability distribution of the affective pulse. In particular, as visually presented in Figure 7, we input the affective pulse as a 6-dimension feature to the recommender algorithm. On the other hand, through early stage measurements, we observe that the social friendship (i.e., the social interaction strength among users in OSN) has a critical impact on the user's tweet click behavior. Then we conduct the social friendship clustering to classify a user's social friends into different groups with different levels of importance. On this basis, we next design a cluster-based Latent Bias Model (LBM) to predict user's likelihood of media click with considering all the above contexts as well as different network and time contexts. Last, by integrating Kaleido into Twidere, an Android Twitter app which has 500,000 downloads on Google Play, we collect user traces from a demographical composition of 16,952 people who consented to report usage data to us. This also enables us to conduct a data-driven measurement and design and realistic experiments to evaluate the performance of Kaleido's mobility support (§5.3). In addition, by collecting system logs from our edge-cloud servers, we reveal the effect of Kaleido testbed (§5.2).

We summarize the major contributions and vantage points of this paper as:

- We collect a large set of real-life mobile traces from 16,952 participants, and reveal the significance and importance of affective feature in social media usage. To our best knowledge, we are the first to employ the affective context in such recommender system.

- We propose a learning-based model to infer affective pulses in media files with 75% accuracy. Furthermore, we design a cluster-based LBM for jointly training affect, behavior, and social contextual features. Through data-driven experiments, we illustrate that it achieves a prediction accuracy of 82%, which outperforms baselines using the same training features.

- We deploy a worldwide edge-cloud testbed for real-life Twidere users by enabling them to offload machine learning procedures with a speed-up of 1,000x factor over the local execution. Our Android prototype consumes user with a low cost of cellular data and energy, which is a significant improvement against the benchmark approaches.

We shall emphasize the feasibility of using Twidere and Android smartphone as a study case. Nevertheless, the proposed methedologies are applicable to most mobile apps and OS platforms.

# 2. KALEIDO: THE FRAMEWORK AND SYSTEM DESIGN

Kaleido provides a real system which consists of an edge-cloud testbed as well as a mobile framework. We start with a brief review of Kaleido system (§2.1), and close the section with describing the key measurements that relate to inspire Kaleido's design details (§2.2).

To keep the illustration concrete, we assume for now that all servers use a learning-based algorithm to identify affective pulses in media; we relax that assumption in the next section. We defer a discussion of the overall algorithm within Kaleido until §4.

## 2.1 System Overview

**Conceptual Framework:** To better present the logic, Figure 2 introduces the framework of Kaleido from a high level. By learning affective pulses, social closeness, and behavior pattern, Kaleido enables users with media recommendation during their operating process. More specifically, when a fresh media content arrives, Kaleido is triggered to take the relevant features of the media context (user behavior, affect, and social friendship) as input to the trained cluster-based LBM, to identify the likelihood of user actions ahead of time. We will elaborate how the proposed cluster-based LBM jointly train the features in §4.3.

**Mobility Support:** Figure 3 depicts how Kaleido mobile framework works in a user-centric manner (i.e., implemented on a user's mobile device), and collects social feeds when accessing new media content with Twitter app. Specifically, it offloads user's machine learning procedure to cloud servers and pushes the training patterns to smartphones asynchronously. Also note that these traces are retrieved using

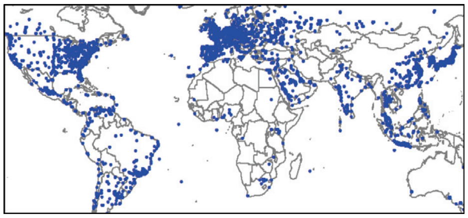


**Figure 4: Kaleido users' demographic composition.**

**Table 1: Description of collected user profiles.**

| Contents | Collected traces |
|---|---|
| Tweets | time, sender, receivers |
| Media files | URL link, sender, receivers |
| User behaviors | publish, like, retweet, mention |
| App usage | launch time, close time, present time |
| Feature training | start time, end time |
| User request | DNS resolution, HTTP metadata |

the Twitter REST APIs [15] in the servers in according to collected user traces. Furthermore, by pushing the training pattern to smartphone, Kaleido executes the inference locally and rapidly. Finally, it highlights the recommended tweets for user in according to the inference results (e.g., see Figure 1).

**System Infrastructure:** Similar to existing techniques, Kaleido employs an edge-cloud architecture for building a testbed to process users' machine learning offloading. Specifically, such architecture supports the time-varying bandwidth and storage allocations requested by different regions. By collecting system logs from early stage deployed servers, we further setup the hardware configuration as illustrated in Table 4. We further discuss the efficiency of such infrastructure in §5.2.

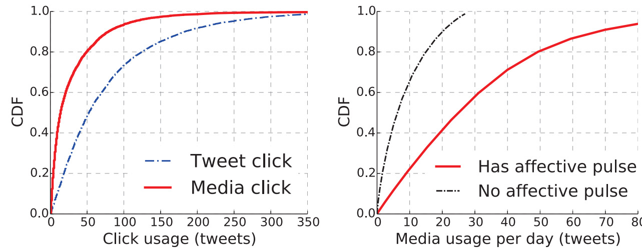
**Caching Policy:** Since the traces are cached temporarily, to ensure privacy, text content in tweets is not recorded and all personal fields are anonymized in advance. In addition, the cached data is uploaded to the cloud server only for further analysis when the smartphone is charging and connected to WiFi. To accelerate the process, Kaleido offloads the machine learning procedure to the testbed whose deployment information are shown in Table 4. Furthermore, the learned patterns are stored on the device to enable real-time inference by Kaleido.

**Network Congestion:** To relief the network congestion caused by the offloading procedure, in this paper, end-user's processing is delivered to the topological nearest server by leveraging her DNS resolution, as we elaborate on in §5.1.

## 2.2 Measurements in Early Stage Work

To capture the effect of affective pulse in recommending social media, we conduct data-driven measurements in early stage work.

Specifically, inspired by a very recent study [14], in this paper, we adopt 6 basic affects in describing human emotional state [16], i.e., *happiness*, *surprise*, *anger*, *disgust*, *fear* and *sadness*, to investigate the forming of users' affective pulses. Furthermore, for evaluating the effectiveness of affective learning in §3, we use a well known dataset from Flickr which covers more than 1 million image traces that




(a) Daily tweets usage (b) Affect-motivated usage

**Figure 5: 16,952 users' app usage across 7 months.**

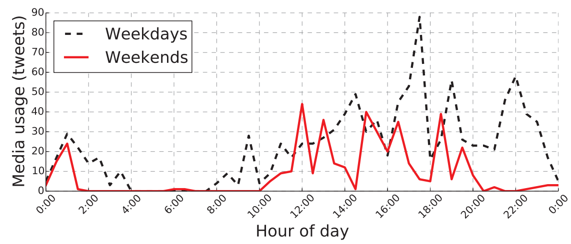



**Figure 6: A user's media usage across hour of day.**

published by 4,725 users and has been manual tagged with the affective labels by prior work [14]. To further investigate users' media behaivor in mobile environment, during the whole early stage, i.e., from March to October 2015, as Figure 4 illustrates, we also have collected data traces from more than 16,952 Twidere [13] users [1] from all over the world with a diverse demographic composition. Because, although Twitter's contents are publicly available, information about when, how, and where they access these social streams are not available in particular in the mobile environment. As the aim is to enable Kaleido system by identifying the affective pulse within media tweets that the user is most interested in, a set of tweet attributes are collected as well. Twidere tracks the user social behaviors (e.g., retweet, like, or mention) of the individual tweets. The source of a tweet is also recorded by identifying whether the tweet is obtained from a direct friend or propagated through friends of others' friends. In addition, with the consent from the user, Twidere enables us to keep track of the user's activity events when reading the tweets (watching, clicking, or commenting along the timeline). The collected trace items are shown in Table 1. Moreover, in this paper, we deploy the Kaleido geographical servers by referring to user composition provided by Figure 4.

The users clicked more than 900 million referred tweets in total. Specifically, as Figure 5 illustrates, on average, a user refreshes 536 tweets a day but only clicked 15% of them. We further observe that more than 60% of the clicked tweets contain media files. Moreover, by referring to the affective learning in §3, we find that more than 76% media have explicitly affective pulses. Note that we infer whether a media file owns an explicitly affect, which shapes the affective pulse, by comparing with the corresponding probability with

[1]Twidere discloses the usage statistics on installation or update. Users are able to opt in or not. There are 43% active users grant us permissions, which indicates user privacy-awareness and the effectiveness of the privacy disclosure. Data are collected in an anonymous style. Also note that the social graphs and tweets are publicly available.

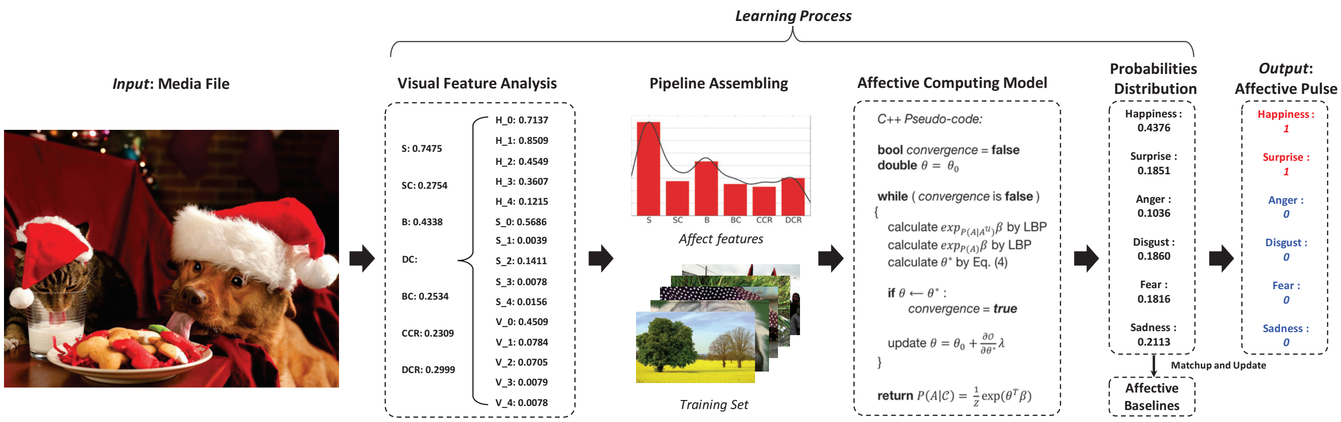



**Figure 7: Flowchart of learning affective pulse from a media. We first extract the values of 7 visual features respectively. By training with the Flickr dataset, it learns the probabilities of 6 affects with the loopy belief propagation (LBP) [17] algorithm. Last, we regularize each $P(a|\mathcal{C})$ (where $a \in \mathcal{A}$) with comparing its probability with corresponding baseline.**

a set of trained baseline thresholds [2] in §3. Figure 6 illustrates that users are time-sensitive, e.g., on weekdays, the user tends to use the app more frequently in the nighttime especially in the midnight, while use the app sporadicly in the daytime. It motivates us to analyze user behavior pattern by taking time feature in accounts. The volume and diversity of data also reflects the real-life behavior of the participant users, which is crucial for understanding and recommending media in mobile social application network traffic, and significant in evaluating the system in a data-driven scheme.

# 3. LEARNING AFFECTIVE PULSE FROM MEDIA CONTENT

In this section, we introduce a learning-based affective computing mechanism by which we identify the affective pulse in a media file.

As aforementioned in §1, the forming of *affective pulse*, e.g., probability distribution of the 6 affects in an object, is complicated. To proceed, we employ a machine learning process. Specifically, we denote the space of affective pulse as $\mathcal{A}$ ={*Happiness, Surprise, Anger, Disgust, Fear, Sadness*}. As different people might have distinct affect even when accessing the same media [3], in our learning algorithm, we adopt the approach provided by [3], i.e., we denote every affect as a linguistic label for the matchup in the learning process. To proceed, we further denote the affect $a \in \mathcal{A}$ expressed by a media file. Inspired by [14], we use visual variables to capture the affect expressed by the media file. We characterize each affect by training with 7 visual features, i.e., saturation (SR), saturation contrast (SRC), brightness (B), bright contrast (BRC), 5 dominant HSV colors (DC), cool color ratio (CCR), and dull color ratio (DCR). Thus, we have feature set $\mathcal{U} = \{SR, SRC, B, BRC, DC, CCR, DCR\}$. For each media $c \in \mathcal{C}$, we have the ground truth $a_c \in \{1, 0\}$ such that $a_c = 1$ ($a_c = 0$) means that whether the media file $c$ has the specific affective feature $a \in \mathcal{A}$ (e.g., happiness) or not. Then we have:

$$f(u_c, a_c) = \frac{1}{z_\theta}\exp\{\theta^T u_c\} \tag{1}$$

where $u_c$ are the mentioned visual features, $a_c$ is the affect express by the media file $c$, $\theta$ is a vector of real valued parameters, and $z_\theta$ is a normalization term to avoid the potential over-fittings. On this basis, the probability distribution $P$ of the affect $a$ in the media files $c \in \mathcal{C}$ can be formulated as:

$$P(A|\mathcal{C}) = \frac{1}{Z}\prod_{\mathcal{C}} f(u_c, a_c) = \frac{1}{Z}\exp\{\theta^T\beta\} \tag{2}$$

where $Z = z_\theta$ is the normalization term, $\beta$ is the aggregation of factor function.

Furthermore, by taking all the affects $a \in \mathcal{A}$ in account, the objective function can be derived as:

$$\begin{aligned} \mathcal{O} = \log P(A|\mathcal{C}) &= \log\sum_{A|A^{\mathcal{U}}}\exp\{\theta^T\beta\} - \log Z \\ &= \log\sum_{A|A^{\mathcal{U}}}\exp\{\theta^T\beta\} - \log\sum_{A}\exp\{\theta^T\beta\} \end{aligned} \tag{3}$$

In addition, the gradient of $\theta$ can be represented as:

$$\begin{aligned} \frac{\partial\mathcal{O}}{\partial\theta} &= \frac{\partial(\log\sum_{A|A^{\mathcal{U}}}\exp\{\theta^T\beta\} - \log\sum_{A}\exp\{\theta^A\beta\})}{\partial\theta} \\ &= (\exp_{P(A|A^{\mathcal{U}})} - \exp_{P(A)})\beta \end{aligned} \tag{4}$$

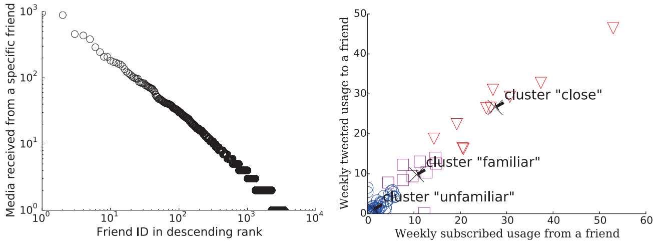
Note that the algorithm updates the parameters by $\theta = \theta_0 + \frac{\partial\mathcal{O}}{\partial\theta}\lambda$. Where $\lambda$ is a regularization parameter to be manually tuned.

**Visual Illustration:** After illustrating the affective learning model, Figure 7 visually illustrates how each visual feature $u_c$ contributes to the proposed mechanism [4]. We take a snapshot of the Christmas cat & dog as a study case, it results in a set of probabilities of {0.43, 0.18, 0.10, 0.18, 0.18, 0.21}. By filtering the insignificant affective features via

[2]In our measurements, the baseline set inclines to {0.24, 0.14, 0.63, 0.19, 0.33, 0.49}. However, it changes with different OSN cases.

[3]Note that since the volume of video sharing in Twitter is rare. Thus, unless otherwise specified, we mainly focus on inferring affective pulses of images.

[4]Note that the DC feature consists of a 15-dimension HSV martix. Due to space limit, we omit the details of DC feature in Figure 7's pipeline assembling illustration, and interested readers can refer to [18].




(a) Her social media usage (b) Her friendship clustering

**Figure 8: Statistics of a user's social interaction frequency and friendship clustering with 1 year usage.**

**Table 2: Media click probability (%) by measuring different features of 16,952 users across one year.**

| | | close | familiar | unfamiliar |
|---|---|---|---|---|
| **Totality** | *Media click* | 51.33 | 28.39 | 13.67 |
| Time Features | *Weekdays* | 56.38 | 32.63 | 13.06 |
| | *Weekends* | 36.94 | 19.30 | 11.75 |
| Interaction Features | *Mentioned by* | 28.57 | 17.72 | 10.81 |
| | *Liked by* | 43.66 | 27.92 | 14.58 |
| | *Retweeted by* | 38.10 | 27.69 | 9.60 |
| | *Replied by* | 51.82 | 28.85 | 13.42 |

comparing with the baselines (as illustrated in §2.2), it reports that the happiness and surprise features are significant and hence have value 1 while the rest affective features are with value 0. Finally, we obtain the affective pulse of {1, 1, 0, 0, 0, 0} as the input affective feature to the recommender model later.

In addition, through training with 80,000 affect-aware images that manual tagged by [14], we further evaluate the accuracy of our learning algorithm. We observe that our model achieves an average prediction accuracy of 0.75, which significantly beyonds the benchmarks in identifying the probability distribution of each affect $a \in \mathcal{A}$ in a specific media. On the other hand, we also discuss the significance and importance of adopting affective pulse in recommending media in §6.

# 4. JOINT RECOMMENDATION WITH AFFECTS AND SOCIAL CONTEXTS

In this section, we first conduct a data-driven analysis on user's mobile behavior and social interactions. Then we reveal their impact on the user's media click actions. On this basis, we last introduce how Kaleido jointly train the affective features with this two features for a socially-driven affect-aware media recommendation.

## 4.1 Behavior Pattern Analysis

In OSNs, the generation and propagation of a media content is simple: any user who generates or re-shares it will become a new host of the media content. Users can fetch these contents from their direct friends in the social network. Intuitively, the social relationships and interactions among a user and her friends have a significant impact on the twittering behaviors. A user might treat different friends differently, and interact with some close friends frequently, while having little contact or response with some unfamiliar friends on Twitter [19].

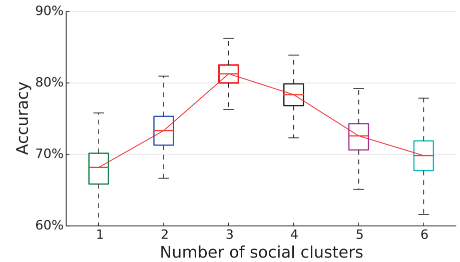



**Figure 9: Accuracy variation of different social cluster number by referring to all users' usage.**

As mentioned in §2.2, user's media click actions enjoy characteristics of high selectiveness and time sensitivity. With this insight, in Figure 8a, we plot the number of one real-life user's clicked tweets with media content (i.e., media tweets) from her friends (i.e., social neighbors) on Twitter in the log-log scale. We rank the set of friends in descending order according to the number of tweets sent by them. We observe a strong power law phenomenon, i.e., almost 70% of the tweets are from only a few friends (less than 10%), and most other friends have little contribution. This demonstrates that friendship (or social interaction strength) plays a critical role on shaping her usage behavior on Twitter.

## 4.2 Social Friendship Closeness

With this insight, we further quantify the impact of social friendship on the user's media tweet click behaviors. To proceed, we first carry out the social friendship clustering. The intuition is that in reality a user typically has very close relationships with a small set of people (e.g., close friends), and is familiar with a group of people (e.g., colleagues). For many other people, the user would have little contact with them. With this observation, we conduct the friendship clustering using the commonly-adopted an unsupervised machine learning with K-Means algorithm [20]. As illustrated in Figure 8b, we utilize the number of tweets subscribed from a specific friend and the number of tweets published by the user to that friend as the features, and cluster the set of her friends into three types: close friends (i.e., cluster "close"), familiar friends (i.e., cluster "familiar"), and acquaintances with infrequent contacts (i.e., cluster "unfamiliar").

After the social friendship clustering, we then explore the impact of friendship on users' media click behavior when accessing the media tweets. Table 2 measures all users' average media click probability under different feature scenarios. In total, we observe that users click the media file with a probability of 0.42 (0.28, 0.13), when the media is sent by a close (familiar, unfamiliar) friend, respectively. This again confirms that social friendship has a significant impact on user's media click behavior. As another example, for the interaction feature, if the media tweet has been replied or mentioned by a close friend, then user would click the media file with a probability gap of 0.17.

**Effectiveness of $K = 3$ Social Clustering:** Figure 9 visually confirms that $K = 3$ achieves a good balance and the best prediction accuracy with a performance gap of 0.13. In addition, it comforts a fact in our daily life observation that people tend to classify their friends into three types: close friends, familiar friends, and acquaintances.

## 4.3 Training with Affect for Recommendation

After illustrating mobile behavior pattern and social friendship closeness, we now introduce the machine learning principle in our system by jointly identifying the set of important training features to build up the learning model.

**Training Context Features:** As mentioned above, we found that three types of context features are critical: affective pulse in media, behavior pattern, and social closeness. Furthermore, we use these three features as the input to our proposed cluster-based machine learning algorithm. For the behavior feature, whenever the user would click a media depends on her regular usage behavior. For the affective feature, what she prefers to click is mainly influenced by each affect that reflect from the media content. For the social feature, the host (close, familiar or unfamiliar friend) of receiving media file is decisive. In the following, we denote the number of these training features as $F$, and their set as $\mathcal{F}$.

**Cluster-Based Latent Bias Model:** We propose the learning model which is based on the Latent Bias Model (LBM) introduced in [21] that aims to utilize proper bias terms to capture the importance of different features for prediction. Here we extend the standard LBM to our case with social friendship clustering, and develop the cluster-based LBM approach for a data-driven learning scheme.

Specifically, we define $b_{f,a,k}$ as the cluster-based bias term to stand for the case that a media is sent by a friend in the friendship cluster $k$, express affects $a_i \in \mathcal{A}$, and contains the feature $f$. Moreover, for a given media content $c$, we first introduce an indicator function $I^c_{f,a,k} \in \{0, 1\}$ such that $I^c_{f,a,k} = 1$ if the media $c$ is sent by a friend in the friendship cluster $k \in \mathcal{K}$ with affect $a \in \mathcal{A}$ and contains the feature $f \in \mathcal{F}$. Then, we define the user action score for the media $c$ as follows:

$$\gamma_c = \alpha + b_0 + \sum_{f\in\mathcal{F}} \sum_{a\in\mathcal{A}} \sum_{k\in\mathcal{K}} b_{f,a,k} I^c_{f,a,k} \tag{5}$$

$\alpha$ is the user's average click rate across all historical media content usage, and $b_0$ is the overall bias for the user. In general, a higher user action score $\gamma_c$ implies a higher probability that the user will click the media content $c$.

Then, the critical task is to train the cluster-based LBM, i.e., to learn the proper bias terms in Equation 5 in order to well capture a user's media click actions. Suppose that the set of historical user data traces (historical media usage set of the user) is denoted as $\mathcal{C}$. For each media $c \in \mathcal{C}$, we have the ground truth $y_c \in \{1, -1\}$ such that $y_c = 1$ ($y_c = -1$) means that the user clicks (open) the media file $c$ in tweet of arrival. To quantify the discrepancy between the prediction based on the media click score $\gamma_c$ and the desired ground-truth output $y_c$, we adopt the widely-used logistic loss measure

$$\mathcal{L}(\gamma_c, y_c) = \log[1 + \exp(-\gamma_c y_c)]. \tag{6}$$

Thus, we learn proper bias terms to minimize the total loss across over the historical data trace $\mathcal{C}$, i.e., $\sum_{c\in\mathcal{C}} \mathcal{L}(\gamma_c, y_c)$. Following common practice in machine learning, in order to avoid overfitting, we also impose $L_2$ regulation into minimization. That is, we minimize the following objective function:

$$\mathcal{O} = \sum_{c\in\mathcal{C}} \mathcal{L}(\gamma_c, y_c) + \lambda \left( ||b_0||^2 + \sum_{f\in\mathcal{F}} \sum_{a\in\mathcal{A}} \sum_{k\in\mathcal{K}} ||b_{f,a,k}||^2 \right), \tag{7}$$

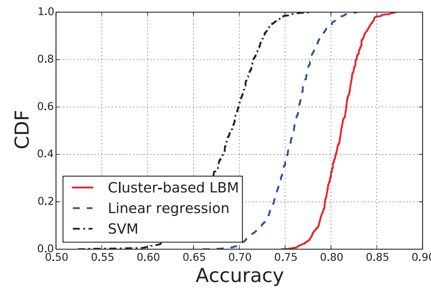



**Figure 10: Accuracy of three baseline algorithms.**

**Table 3: Summarization of Figure 10.**

| Algorithm | Balanced accuracy | Best accuracy | Worst accuracy |
|---|---|---|---|
| Cluster-based LBM | 0.82 | 0.87 | 0.73 |
| Linear regression | 0.71 | 0.82 | 0.65 |
| SVM | 0.68 | 0.77 | 0.52 |

where $\lambda$ is the regularization parameter to be manually tuned.

Since the function in Equation 7 is convex, we can apply the first-order condition and derive the gradients as

$$\frac{\partial \mathcal{O}}{\partial b_0} = -\sum_{c\in\mathcal{C}} \left( \frac{\exp(\gamma_c y_c)}{1 + \exp(\gamma_c y_c)} \right) y_c + 2\lambda b_0, \tag{8}$$

$$\frac{\partial \mathcal{O}}{\partial b_{f,a,k}} = -\sum_{c\in\mathcal{C}} \left( \frac{\exp(\gamma_c y_c)}{1 + \exp(\gamma_c y_c)} \right) y_c I^c_{f,a,k} + 2\lambda b_{f,a,k}. \tag{9}$$

Similar to many machine learning studies, we can adopt the Stochastic Gradient Descent (SGD) method [22], to learn optimal bias terms. The key idea is to utilize the data samples to iteratively update the gradient as follows:

$$b^{t+1}_* = b^t_* - \epsilon_t \frac{\partial \mathcal{O}}{\partial b^t_*}, \tag{10}$$

where $b^t_*$ denotes a given bias term at the $t$-th iteration and $0 < \epsilon_t < 1$ is the smoothing factor for updating. As shown in [22], the SGD method converges to the optimal learning point provided a sufficiently small $\epsilon_t$.

After learning, when a fresh tweet with media $c$ arrives, we predict a user's click likelihood using the loss measure in (6). Specifically, we predict that a user will click the media file if $y_c = 1$ has a lower risk, i.e., $\mathcal{L}(\gamma_c, y_c = 1) < \mathcal{L}(\gamma_c, y_c = -1)$. In this case, the media is clicked by the user and otherwise in reverse.

**Model Baselines:** We last evaluate the performance of the proposed cluster-based LBM algorithm by referring to all the participants' usages. Figure 10 depicts the cumulative probability distribution (CDF) of the prediction accuracy for all tested tweets. As mentioned above, when the friendship cluster number $K = 3$, it achieves the best performance with an average prediction accuracy of 0.82. As baselines, we also evaluate the prediction process with linear regression (LR) in [11] and SVM. Jointed with Table 3, Figure 10 evaluates all baselines' performance that LR approach can mostly achieve an average prediction accuracy of 0.71, which outperforms SVM approach by 3%, but be inferior to the cluster-based LBM by 11%. This demonstrates the efficiency of the cluster-based LBM. The gain of cluster-based LBM stems from the fact that the defined cluster-based bias

Table 4: Details of network topology, geographic distribution and configuration of Kaleido testbed.

| Region | Location | Carrier and OS | Bandwidth | Service | Configuration (CPU, Memory, Storage) |
|---|---|---|---|---|---|
| US | Los Angeles | Vultr (Debian 8.0) | 10Mb | Server-load Balancing | 2.4GHz single-core, 768MB, 20GB |
| | San Mateo | Aliyun (Debian 7.5) | 5MB | Caching, Content Services | 2.3GHz dual-core, 2GB, 1TB |
| Europe | Frankfurt | Vultr (Debian 8.0) | 10Mb | Server-load Balancing | 2.4GHz single-core, 768MB, 15GB |
| Asia | Tokyo | Vultr (Ubuntu 14.04) | 10Mb | Server-load Balancing | 2.4GHz single-core, 512MB, 20GB |
| | Shanghai | Tencent (CentOS 7) | 5MB | Caching, Content Services | 2.6GHz quad-core, 12GB, 1TB |
| | Beijing | Tsinghua (Debian 8.0) | 1MB | Caching | 2.2GHz octa-core, 128GB, 6TB |

terms can well capture the impact of affective feature on user's media click, we elaborate it in §6.

# 5. EXPERIMENTS

In this section, we conduct the trace-driven evaluation over the testbed and Android smartphones to investigate the performance of Kaleido.

## 5.1 Implementations

To evaluate the performance of Kaleido system, we deploy a worldwide network testbed to undertake users' machine learning procedure. Table 4 summarizes the network topology, geographic distribution and configuration of our testbed. Specifically, there are 3 categories of service for the deployed servers, i.e., server-load balancing (as a streaming server), caching (as a data center), and content service (as a computing center).

Moreover, to illustrate the effect of performance of the provided mobile framework, we also run a trace-driven evaluation for the 16,952 users that each of them keeps active for a long and consecutive period of at least 3 months with detailed trace records. The emulator runs on a Google Nexus 6, Nexus 5 and Samsung Galaxy S4 smartphone, respectively, that with access to both China Mobile TDD LTE network as well as a campus WiFi network. The emulator reads and replays the usage events collected from real-life users, including connecting to or disconnecting from WiFi networks, accessing Twitter, and reacting its media files.

## 5.2 Testbed Measurements

The testbed serves a diverse workload spanning massive machine learning procedure: friendship clustering, training, media content downloading, content caching, and other miscellaneous process. We use the latest 3 months' (from December 2015 to March 2016) user request logs and UNIX round robin database (RRD) logs collected from 6 vantage servers, i.e., 2 in US, 1 in Europe and 3 in Asia, to measure the performance of our deployed testbed. To this end, we leverage the Metalink standard [23], which is an XML-based download description format that provides the metadata of the content [5]. Metalink-enabled HTTP clients and proxies understand the relevant HTTP headers (e.g., to verify the authenticity and integrity of the data, discover faster mirrors, etc.), while legacy clients simply ignore them. In addition, the RRD log reports the system load, network bandwidth and stock prices with a constant disk footprint.

Table 5 summarizes that the balance and costs of Kaleido edge-cloud infrastructure. Specifically, we recorded that the most heavy system load appeared in the machine learning process server (with an average usage of 21.4%) while the most bandwidth consuming was the server-load balancing (with an average usage of 0.2MBps) [6]. In addition, the latency for all the Kaleido edge servers are less than 670ms which again comforts the factuality of our testbed.

Table 5: Testbed performance across 3 months.

| Service | Latency (ms) | Bandwidth usage (%) | System load (%) |
|---|---|---|---|
| Service | 170 | 4.7 | 21.4 |
| Caching | 211 | 3 | 7.3 |
| Server-load | 670 | 14.4 | 5.7 |

## 5.3 Benchmarks

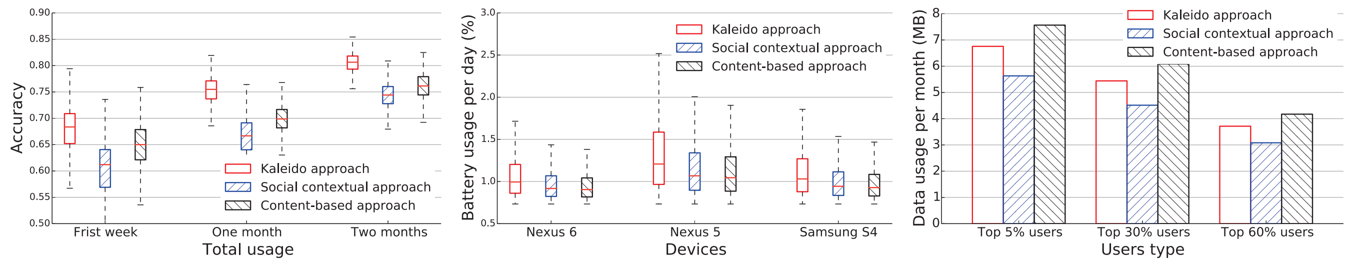
After the testbed efficiency has been discussed, we next evaluate the performance of Kaleido mobile framework running on the smartphones in against with existing representative recommendation algorithms. As illustrated in Figure 10 the cluster-based LBM algorithm in Kaleido is very efficient, and achieves the average prediction accuracy of 0.82. Upon comparison, the linear regression (LR) algorithm using tweet training features can only achieve the average prediction accuracy of 0.73. In the following, we also compare different recommendation algorithms given as: In the following, we further consider the three recommendation approaches as performance benchmarks in this paper as:

- *Kaleido approach*: We implement the recommender system by running the proposed cluster-based LBM with affective context, user behavior contexts, and social contexts.
- *Content-based approach*: We implement the recommender system with content training features in [6].
- *Social contextual approach*: We replace the content features with social information contexts in [12].

Specifically, we evaluate the performance of Kaleido recommender system, by considering different users, i.e., top 5% users (the most 750 active users who refresh 1,301 tweets and consume 40 media tweets per day on average), top 30% users (users that daily refresh 780 tweets and consume 22 media tweets), and top 60% users (users that daily refresh 429 tweets and consume 11 media tweets). For the evaluation of each approach, the same user trace are replayed, and

[5]E.g., see http://releases.ubuntu.com/releases/15.10/ubuntu-15.10-desktop-i386.metalink.

[6]Note that this low average bandwidth consumption is due to the training procedure for all users are sporadicly triggered when each smartphone is in charge and WiFi available and after a time slot of one day from the last calling. In addition, user profiles are usually no more than 200KB.




**(a)** Accuracy growth across time **(b)** Battery overhead for using Kaleido **(c)** Data usage for using Kaleido

**Figure 11: Benchmarks across accuracy growth, daily battery overhead and monthly cellular data cost.**

media files are obtained by using the WebView component provided by the Android framework. To embrace stable data and battery consumption results, we evaluate this two items by training with each user's first 2 month trace and using at least another 2 month traces for testing.

**Accuracy Growth:** We first compare different approaches' accuracy growth with time. As illustrated in Figure 11a, our algorithm tends to be stale after one month's usage. We see that Kaleido approach climbs from 0.67 to 0.82 when after 2 months' use. Upon comparison, content-based (social contextual) approach starts with lower accuracy of 0.62 (0.65) and reaches to a balanced accuracy of 0.74 (0.76). This is due to the fact that affective feature plays a significant role in learning users' media click (we shall discuss it in §6). This demonstrates the efficiency of the proposed Kaleido approach in media recommendation.

**Data and Battery Overheads:** We next compare different recommendation approaches in terms of cellular data consumption of Twidere app (which includes the cellular traffic overhead by both Kaleido and on-demand contexts fetching by the user), and battery consumption of the app per day (i.e., the percentage of the fully-charged battery capacity). The results are shown in Figure 11c and Figure 11b respectively. We observe, take the top 5% users for instance, that Kaleido uses 6.8MB cellular traffic per month and about 1.4% battery usage per day on average. On the other hand, the content-based (social contextual) approach costs users with 7.5MB (5.8MB) cellular traffic per month and 1.1% (1.2%) battery usage per day on average.

Moreover, evaluations in Figure 11 also illustrate Kaleido outperforms existing approaches in all kinds of usage cases.

# 6. DISCUSSIONS

**Does Affective Pulse Really Matter?** Kaleido, as mentioned above, is the first step towards an affective media recommender system. Performance of the proposed algorithm in this paper demonstrates that Kaleido is promising when integrating the affective feature with user behavior and social friendship contexts. However, we have yet known how critical the affective feature plays. To understand it, in Figure 12, we compare the accuracies of all the mentioned algorithms in §4.3 (with the same user group) by removing the affective feature, i.e., we only put the behavior and social contexts into the training model. Similar to the prediction evaluation above, we adopt the $K = 3$ social clustering as the study case. We observe that the LBM enjoys a positive impact with affective feature and achieves an average prediction accuracy of 0.71, which has 11% performance

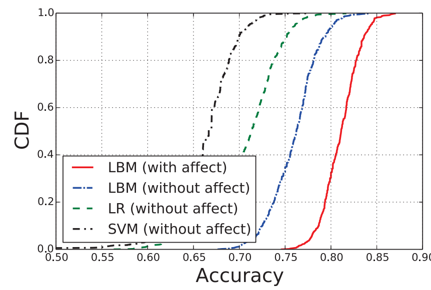
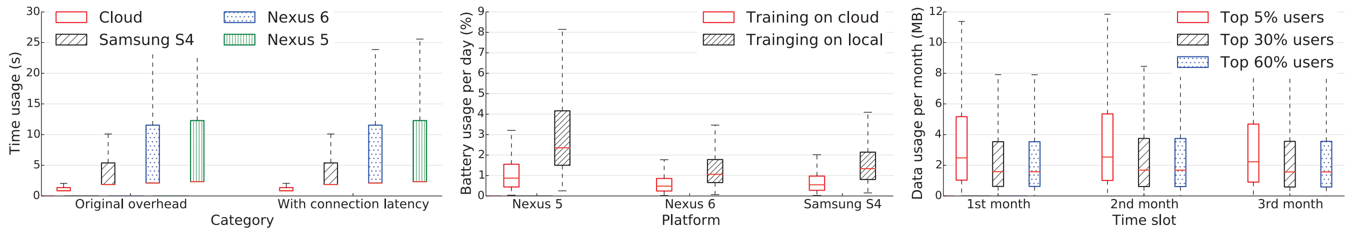



**Figure 12: Evaluation of no affective feature in cluster-based LBM (11% degradation), LR (3% degradation), and SVM (2% degradation).**

degradation with respect to the standard algorithm. Upon comparison, the LR (SVM) algorithm achieves an accuracy of 0.68 (0.66), with a slightly small performance degradation of 0.03 (0.02). This again demonstrates the uniqueness and significance of Kaleido approach in exploiting the affective feature for efficient media recommendation.

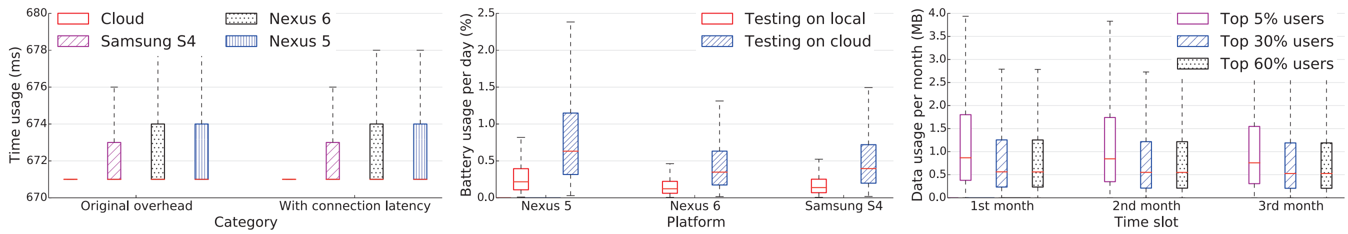
**Why Training on Cloud?** One key component of Kaleido is to implement the cluster-based LBM algorithm for the data training (i.e., learning the optimal bias terms from the user data traces). Intuitively, there are two data processing approaches: 1) processing on local device, i.e., we conduct the data training procedure on user's smartphone locally. 2) processing on cloud, i.e., we offload the data training to the cloud server is to leverage the strong parallel computing power to speed up the data training. To investigate the characteristics, in Figure 13, we emulate the average time overhead of these two data processing approaches with the top 60% active users, by setting different size of user trace in the learning algorithm, with considering the network connection latency we measured in Table 5. We find that the cloud approach can significantly decrease the daily time overhead for data processing, by a factor of 1,000. We further compare the monthly data consumption for training on cloud for different users respectively. We compare that this procedure consumes user largely 2.5MB [7] data per month, which confirms that the effectiveness of training on cloud. Note that since the user's behavior tends to be stable, to further save both cellular data and energy usage, we can aggregate the training data for a longer while (e.g., one week) but not everyday any more, and carry out data training weekly by

[7]Kaleido consumes additional 4.5MB/month since additional data/information download is needed for the testing.




**(a)** Daily training time consumption **(b)** Daily battery overhead for training **(c)** Monthly data cost for training

**Figure 13: Overhead with different training schemes for the machine learning procedure in Kaleido.**




**(a)** Daily testing time consumption **(b)** Daily battery cost for testing **(c)** Monthly data consumption for testing

**Figure 14: Daily costs with different testing schemes for Kaleido.**

offloading the training to the cloud only when the user is on WiFi while charging.

**Why Testing on Local?** Furthermore, we also evaluate the testing schemes based on above approaches. The results are shown in Figure 14. We observe that the cloud approach consumes more energy (0.5% of total battery usage per day), leading to additional cellular data usage (0.9MB per month). In addition, since the cloud approach requires network connection, it brings a latency of 670ms each time, daily time overhead for testing all the tweets on local ties to the cloud approach. Thus, the local testing approach is more preferable. The reason is that, different from the training process that is computation-intensive, the testing procedure is data-centric and offloading to the cloud would incur higher delay and energy overhead for data exchange between the cloud and the device.

# 7. RELATED WORK

In this section, we review three directions of prior research directly related to our work. Specifically, we highlight the key differences of Kaleido against them respectively.

**Media-based Affective Computing:** Although many recent studies, e.g., [24], keep on emphasizing the significance of affective media computing, but their approaches are still highly subjective and difficult to embrace quantitative measurements [25]. A recent study [14] focus on training data and models for identifying the emotional influence, based on the ground-truth affects that are manually labeled in order to guarantee the prediction accuracy. However, it also brings a challenge to efficiently process massive images in OSNs [14]. Along a different line, motivated by the insight that user behavior, social friendship and media affect play critical roles on user's emotion-triggered action in OSNs, in this paper, we propose a novel learning-based mechanism to intelligently deal with media recommender system which support the affective-aware recommendation.

**Recommendation Techniques:** There are two prevalent schemes for building recommender systems, i.e., content-based (CB) [11] and collaborative filtering (CF) [26]. The CB method is on a basis of recommending items, e.g., images or videos, that are similar to those in which users are interested in according to the historical feeds. The CF approach, on the other hand, recommends items to the user based on other individuals with similar preferences or tastes. Many recent studies, such as [12, 27], are built on both CB and CF systems, usually by rating a set of items. To avoid this extra burden on the user, leveraging implicit interest indicators [28], such as the purchase history, views, clicks, or queries, has recently become more popular in recommender systems. Motivated by the insight that time, social, and network context play critical roles on users' media click behavior, in this paper we propose a novel recommendation system based on the generalized cluster-based bias model.

**Mobile OSN Studies:** To analyze social behaviors of mobile Twitter users, [29] identifies people using microblogging to talk about their daily activities and to seek or share information as well as analyzing the user intentions associated at a community level, showing how users with similar intentions connect with each other. In addition, a number of recent studies, such as [30, 31], address the problem of computing influence in Twitter-like networks and finding leaders whose tweets are influential. Our work does not aim at finding users who are influential directly. Instead, we exploit that different social friends have different impact on a user's activation on media usage or propogation. [32] proposes a tree-based algorithm to mine user-friend graphs to discover strong friends of a user. In contrast to our work, [32] does not consider how to utilize the social friendship structure to facilitate the information and content sharing among users, in particular, under a rich communication environment.

## 8. CONCLUSION

We have presented Kaleido, a system for smart OSN media recommendation on smartphones. Building upon our cluster-based machine learning mechanism, Kaleido automatically learns relationships among various content and context impacts. Experiments with real Twitter traces from 16,952 people and an Android prototype show that Kaleido can achieve superior performance of a significant media recommendation accuracy while with minor additional data or energy consumption. Moreover, our design enables offloading of machine learning procedures to a cloud server, and achieves a speed-up of up to about 1,000 over local execution on smartphones. For future work, we will consider extending our system with a comprehensive implementation to support more media formats, e.g., video.

## 9. ACKNOWLEDGMENTS

We thank the Multimedia reviewers for the helpful comments on this paper, Ningyuan Li, Liyuan Wang, Boya Wu, Hangcheng Zhao, Taoran Tang and Yaohua Pu for preliminaries. We also thank all the data providers for their contributions on this paper. This work is supported by Tsinghua University Initiative Scientific Research Program, National Key Research and Development Plan (2016YFB1001200), National Basic Research Program (973 Program) of China (2012CB316401), and National Natural, and Science Foundation of China (61370023).

## 10. REFERENCES

[1] Arjan Geven, Manfred Tscheligi, and Lucas Noldus. Measuring mobile emotions: Measuring the impossible? In *MobileHCI*, page 109. ACM, 2009.

[2] Marco Guerini and Jacopo Staiano. Deep feelings: A massive cross-lingual study on the relation between emotions and virality. In *WWW*, pages 299–305. ACM, 2015.

[3] Jie Tang et al. Quantitative study of individual emotional states in social networks. *Affective Computing, IEEE Transactions on*, 3(2):132–144, 2012.

[4] Rosalind W Picard and Roalind Picard. *Affective computing*, volume 252. MIT press Cambridge, 1997.

[5] Lisa Sayegh et al. Managerial decision-making under crisis: The role of emotion in an intuitive decision process. *Human Resource Management Review*, 14(2):179–199, 2004.

[6] Nikolaos Georgis et al. System and method of selective media content access through a recommednation engine, August 17 2007. US Patent App. 11/840,814.

[7] Elizabeth A Vandewater et al. Digital childhood: electronic media and technology use among infants, toddlers, and preschoolers. *Pediatrics*, 119(5):e1006–e1015, 2007.

[8] Börkur Sigurbjörnsson and Roelof Van Zwol. Flickr tag recommendation based on collective knowledge. In *WWW*, pages 327–336. ACM, 2008.

[9] Martin Prangl et al. A framework for utility-based multimedia adaptation. *Circuits and Systems for Video Technology, IEEE Transactions on*, 17(6):719–728, 2007.

[10] Moon-Hee Park et al. Location-based recommendation system using bayesian user's preference model in mobile devices. In *Ubiquitous Intelligence and Computing*, pages 1130–1139. Springer, 2007.

[11] Michael J Pazzani and Daniel Billsus. Content-based recommendation systems. In *The adaptive web*, pages 325–341. Springer, 2007.

[12] Meng Jiang et al. Social contextual recommendation. In *CIKM*, pages 45–54. ACM, 2012.

[13] Twidere-twitter client. http://twidere.mariotaku.org.

[14] Xiaohui Wang et al. Modeling emotion influence in image social networks. *Affective Computing, IEEE Transactions on*, 6(3):286–297, 2015.

[15] Twitter rest apis. https://dev.twitter.com/rest/public.

[16] Paul Ekman. An argument for basic emotions. *Cognition & emotion*, 6(3-4):169–200, 1992.

[17] Kevin P Murphy, Yair Weiss, and Michael I Jordan. Loopy belief propagation for approximate inference: An empirical study. In *Uncertainty in artificial intelligence*, pages 467–475. Morgan Kaufmann Publishers Inc., 1999.

[18] Costantino Grana et al. Enhancing hsv histograms with achromatic points detection for video retrieval. In *CIVR*, pages 302–308. ACM, 2007.

[19] Mark EJ Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103(23):8577–8582, 2006.

[20] Paul S Bradley and Usama M Fayyad. Refining initial points for k-means clustering. In *ICML*, volume 98, pages 91–99. Citeseer, 1998.

[21] Liangjie Hong et al. Learning to rank social update streams. In *SIGIR*, pages 651–660. ACM, 2012.

[22] Léon Bottou. Stochastic gradient descent tricks. In *Neural Networks: Tricks of the Trade*, pages 421–436. Springer, 2012.

[23] Anthony Bryan et al. Metalink/http: Mirrors and hashes. Technical report, 2011.

[24] Bing Li et al. Context-aware affective images classification based on bilayer sparse representation. In *MM*, pages 721–724. ACM, 2012.

[25] Marco Cristani et al. Unveiling the multimedia unconscious: Implicit cognitive processes and multimedia content analysis. In *MM*, pages 213–222. ACM, 2013.

[26] David Goldberg et al. Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35(12):61–70, 1992.

[27] Ido Guy et al. Social media recommendation based on people and tags. In *SIGIR*, pages 194–201. ACM, 2010.

[28] Mark Claypool et al. Implicit interest indicators. In *IUI*, pages 33–40. ACM, 2001.

[29] Akshay Java et al. Why we twitter: understanding microblogging usage and communities. In *SNA-KDD*, pages 56–65. ACM, 2007.

[30] Meeyoung Cha et al. Measuring user influence in twitter: The million follower fallacy. *ICWSM*, 10(10-17):30, 2010.

[31] Chao Wu et al. Spice: Socially-driven learning-based mobile media prefetching. In *INFOCOM*. IEEE, 2016.

[32] Juan J Cameron et al. Finding strong groups of friends among friends in social networks. In *DASC*, pages 824–831. IEEE, 2011.