Application Recommendation Service using Collaborative-Filtering in a Mobile Cloud Environment

1Silviya Nancy J and 2Udhayakumar S
1Assistant Professor, Department of Computer Science and Engineering, Rajalakshmi Engineering College, Thandalam, Tamilnadu, India
2Associate Professor, Department of Computer Science and Engineering, Rajalakshmi Engineering College, Thandalam, Tamilnadu, India

ABSTRACT

The progression in the market of cloud services deployment for major sectors like education, research, industry, health care, government, military, etc., has reached its edge with the paradigm rise of “everything and anything as-a-service”. Moreover, the augmented usage and ubiquitous nature of smart phones have shaped a huge expectation on service-orientation which in turn fueled the custom of fingerling the services in the pocket instantly on demand. Even though the application prospects are met straightaway through handheld devices, there are certain constraints like storage, processing, and computation inadequacy. To overcome these restraints, Mobile Cloud Computing (MCC), offers a pervasive platform for processing data and computation-intensive applications. This study is proposed to build the recommendation system using the collaborative-filtering algorithm, which will be identifying the user context based on the current situation with the analytical classification of past experiences for management of mobile devices users and their services. The system collects the information about the user and the data-intensive service that is needed by the user. This would make an effective offloading decision for the users to provide the best service. Along with that, the ideal use of predictive algorithms enables the recommendation engine to offer the users with an additional list of pre-defined services which would provide the consumers with better insight on service usage.

Key words: Cloud computing, Mobile cloud computing, Item-based Collaborative-filtering, Predictive algorithm, Recommendation engine.

INTRODUCTION

We actively exist in the phase of cutting-edge technologies of mobile computing. With the progression of technical advancements starting from featured phones to smart devices have driven the consumers to greater extend which has envisioned to create a hype by advocating and employing any type of service in the users’ pocket. The increased growth has reached great propaganda with the evolutionary developments like tablets, where the handheld devices are becoming smarter, that out stands the ability of a user itself. The smart devices of this era have marked their presence with their incredible features like internet, scalability, portability, accessibility, availability, network access with wider bandwidth, etc. This has also increased the marketing strategy of smart devices as well as usage of services online offering service-oriented architectural features for accessing any kind of applications. The mobile users are also constantly increasing with the exemplary
features. A brief survey on smart devices by International Data Corporation (IDC) has forecasted that the growth of smart mobile devices in 2016 is around 2.5%, which will exceed in 2017 to the range of 4.2% which is 1.53 billion and it will be 1.77 billion in 2021.

The striking growth of mobile devices targeted the market strategies and a hike of cloud services and vice versa. Of course, cloud computing has already sparked its significance by revolutionizing the delivery of on-demand services in all possible ways which instigated even the featured mobile users to change to smart devices to handle the service-oriented web services eloquently. To elaborate the concept of cloud computing concisely, “on-demand services – anywhere, anytime, with internet”. It inculcates the conceptual ideologies of service-oriented and web services architecture along with the prospective evolution of Virtualization. The collaborative use of above mentioned technical concepts has paid a great heed to the under-utilized resource, which offers a platform for exploiting to-be-extinct resources properly. Already many traditional install-and-use softwares’ has already been shifted to the cloud, which is in turn provided as a service to the users marking its stay. Some the well-known offered services include Microsoft 365, Adobe Creative Cloud, Amazon Web Services, Google Cloud Platform. Nevertheless, these vendors have penetrated the market by increasing the usage of cloud services through smart devices which have gradually exceeded the use of PCs and Desktops.

Even though the smart devices have all the exciting features, it has few physical inadequacies like limited storage, computational processing capacity and restrained battery power, the use of virtualized resources online is restricted and constrained. This is where the Mobile Cloud Computing (MCC) era started by computing the intensive mobile applications in cloud space. To quote a few computationally exhaustive applications include handling of big data in form of images, documents, digital and video contents, etc. It is manipulated by offloading the huge data into the cloud space from mobile to cloud server that can handle the huge size of data. So, by now the smart devices require a middleware for efficient transfer of highly computational intensive tasks which is substituted by WiFi and WiMax. This strategy of offloading has become more popular among the cloud consumers, where the vendors are more affluent in offering it.

Figure 1 Cloudlet Representation with Mobile Cloud Computing (MCC)
Context plays a vital role in developing and managing any type user-dependent applications. To precise the idea of context, it is a portrayal of an environment and the behavior of an object (person or place), which in turn depicts the interactive relation of that particular object with the environment and the application to which the object is currently adhered to. The seamless strategy of context will enable better service delivery by increasing the adoption, access, with respect to the offered contextual information. This also includes the situation-awareness which percepts the evidence about the current scenario, which will also extend a platform for classification and prediction. The main reason for the existence of context awareness is so that the system can perceive its surrounding, as close as possible to the users. The system has to understand what the user needs without the user explicitly stating it, with minimum context mismatch.

The most important aspect in offloading computation to the server is context-awareness. It is because the contextual factors play a vital role in consumers’ preferences. When it is claimed that users’ predilection is additionally essential in managing and delivering a service, the context of the user plays a major role, which also compiles their interest and affliction. In recent years there has been an extensive research focus is on context-awareness, this is because the traditional recommender systems did not consider users’ preferences based on context but only the history.

In this paper, we propose a context-aware collaborative-filtering algorithm. The working functionalities predict the users’ perception and context based on the prior analytics of experience and priority. This is done by grouping the users based on the statistics of the previous context. The impediment arises on performance measures and overhead when multiple users try to access the same service or different service at the same instance. The solution to this overhead is critically given by the conceptual theories of Artificial Intelligence (AI). The field of AI is a fastest adapted technology that embodies the predicament in all the aspects especially in users’ preferences and tasks. The classification and prediction can be executed by the Machine Learning (ML) which is been derived from AI have augmented a characteristic elucidation by embedding a Recommendation Engine, which manages the users and their distribution while using services from cloudlet when many users target it.

The organization of the paper includes is depicted as follows, the Chapter 2 depicts a deep survey of various literature and articles on cloud computing, mobile cloud computing and offloading, context-awareness and machine learning. Chapter 3 elaborates the proposed recommendation engine with collaborative-filtering and Chapter 4 reviews on results and discussions and Chapter 5 ends with a conclusion and future references.

REVIEW OF LITERATURE

Review on Mobile Cloud Offloading

The rise and shift in mobile cloud technologies have emphasized a great space in the field of information technology, education, and research. The mainstay of this chapter is to portray the different reviews of the literature. The research articles from various authors have accentuated the familiarity of offloading services on to cloud from smart devices. The following papers of discusses the necessity of the
proposed concept.

The author Mahbub E Khoda et al (2006), presented code offloading system which is intelligent enough to handle efficient decisions over 5G networks. The article proposes the metrics of performance evaluation in terms of saving computational power and energy. Khadija Akherif et al (2016) explains the demanding expectations on Cloud and Mobile Cloud Computing. The presenters have enlightened the need and reasons for offloading such as limited battery life, processing, and limited storage. They have also depicted the issues and challenges in smart devices while executing computational intensive tasks, thereby clarifying the security and privacy applications of mobile cloud applications. The issues are addressed in various phases like smart handheld devices, cloud data centers and transmission of computational files over the network. The proposal directly insists on secure computation offloading of linear programming. It also underlines various offloading frameworks in the light.

The following paper presents Mobile Augmentation Cloud Services (MACS), a middleware proposed by Dejan Kovachev and Ralf Klamma (2012). They state an affluent offloading decision of mobile application from Android device by adaptive partitioning, resource scheduling and monitoring by transparently exhibiting the offloading procedure. The implementation also claims that around 95% of battery is gained in offline when compared to local execution. Monika Dudeja and Kritika Soni (2014), the authors encapsulated a brief survey on the necessity of offloading applications on to the cloud for processing. They have done an experimental analysis on a game with the time taken by offline and online execution and processed the result saying that offloading of intensive tasks saves energy when compared to offline execution on the device itself.

Chien-Hung Chen et al (2016) in their article on Mobile Cloud Framework for Deep Learning enlightens the powerful embellishment of Deep Learning and cloud computing which is been in turn emerged into Mobile Cloud Computing in Deep Learning. The authors highlight the emergence of huge data which cannot be handled by smart devices due to constrained storage and processing power. The model used smart car camera that detects the objects which are implemented on NVIDIA Jetson TK1. Zhaosheng Zhang, Shuyu Li (2016), the writers of this article gives a brief view on existing research on offloading decisions and comparison of different frameworks along with the challenges. Bowen Zhou et al (2015), have an offered a vibrant explanation for the prime emergent of the mobile cloud environment. It has a precise recommendation on offloading decisions depending on the context of the users’ device and environment.

Reviews on Interaction System based on Context-Awareness and Collaborative-filtering

The next section of literature review focuses on interaction/recommendation system for offloading clients. This comes up the scenario that has a main emphasis on user interaction and satisfaction when using the service.

The writers of the article Nagarajan Natarajan et al (2013), highlights the collaborative filtering with the context of user interaction. The paper has done elaborative findings on the applications used by the smartphone users based on the context. They have provided a brief explanation of recommender systems that are used to identify consumers’ interest based on the context that includes their current location, likeness, and season. To enlighten this, the team of authors has proposed an iConRank algorithm which clusters the users
based on their behavior and context whose main objective is personalization.

Linas Baltrunas and Francesco Ricci have presented (2009) a pre-filtering technique for context-aware recommendation systems. Based on the ratings that are given for the application, it is split according to the users’ context. This is where the item splitting is been done and different groups of ratings are elaborated. This enables to understand the users’ interest. Linas Baltrunas and Xavier Amatriain (2009), the authors have presented the pre-filtering techniques based on implicit user feedback. They have split the individual user profile based on the requirements keeping their context as a primary concern. The feedback is maintained as a single container and evaluated on time-aware music recommendation system.

The review of literature focuses on exemplary explanations on various aspects of Mobile cloud computing (MCC) and context-awareness by inculcating various users’ context. The survey provides a platform to understand and build recommendation engine with different requirements. Offloading computation to the server has the following requirements as follows, checks whether the mobile processor will be able to handle the task, if so how long will it take to complete the task. Simultaneously also rectifies the power consumption, network bandwidth availability needed for computation in mobile itself. If these conditions are not met and when there is overhead, the intensive task must be shifted to the resource-rich server for computation, processing, and storage. With respect to the application, the consumers’ context also becomes primarily important for the efficient marketing of applications. The following have to be considered, initially identify the client feasibility and requirements by thoroughly concentrating on users’ identity and behavior that includes the capability of the devices and importantly the location. With these insights, the recommendation system is designed and manipulated in the following chapter.

PROPOSED METHOD

The volume of information accessed and created by the user is growing rapidly with the adverse effect of technological development. The sources of information are available in many different paradigms starting from our daily activities to books and exclusively through ever developing the web. At all these junctures, context and situation occupy a foremost place. Understanding the users’ perspective has become the criteria for any business and marketing. Many inventions have been forecasted to adhere and master the users in recent years with the development of recommender systems, where these engines are monitored and controlled by mobile devices. These devices take an at most effort by manipulating the user environment, context, location, and surrounding including the device viability. For instance, if the user is located in public places like shopping malls, railway stations, or at home or even workplace, the application serves the user according to the context by taking care of user environment and device requirements with the higher priority of users’ interest. To cater all these ideas, many recommendation algorithms were introduced. One of the efficient algorithms to handle users’ preferences and recommendations is a collaborative-filtering algorithm. The proposed system initiates by evaluating the interests of the users with the former history. For instance, if a user is requesting for the service, the recommendation engine would analyze the prior login attempts and service usage of each and
every user individual user. The proposed system embodies the following characteristics which is also depicted in Figure 2,

1) Context-based Code offloading mechanism for intensive tasks. The intensive tasks include Video conversion, audio conversion, object recognition, book detection, face recognition, video analytics, etc.

2) The above mentioned intensive tasks are made available to the user based on the users’ preference and priority through the frequent-item set algorithm (item-based CF) with analytics on the previous history based on various types of filtering methodology.

3) Design of recommendation engine based on a collaborative-filtering algorithm.

Having discussed adequately on the technical aspects of code offloading, the focus of the proposed system is intended with the development of Recommendation System grounded on the collaborative-filtering algorithm. In general there are two different approaches to filtering depicted as follows, content based approach and item-based and user-based collaborative filtering (CF). CF takes the ratings of the user on similar behaviour and likeness. The primary idea of collaborative filtering is to identify the individual’s opinions that provide the recommendations on items with which prediction can be done with explicit opinions obtained from consumers. The content-based approach needs extended acquaintance and understanding of the applied items. So the former is best suited for this proposed architecture.

The collaborative filtering can also be referred as collective-filtering, which filters the information using the recommendations of other people. The filtering is based on assessing the likeness of the users’ interest from any set of items of scenarios like music, books, apps, etc. It is centered on the hint that users’ who have agreed to the assessment of those definite items in the past and likely to agree again in the future. The most common
example for collaborative-filtering under the category of item-based filtering is a rating of a movie. It is premeditated by prediction of the active user by weighting the average ratings of the selected users. As mentioned earlier, the proposed system is segmented into the following modules that include the sequence drawn in Figure 3,

a) Applications in the cloudlet with user rating based on various categories.

b) Computation offloading by the user and selecting applications based on the prime ratings handled by recommendation engine centered on the context of the user.

c) Execution of task and storage in the cloud server.

RESULTS

As discussed earlier in the proposed system, centered on the context attributes users decides to offload the intensive computation to the resource rich server. The two applications that are taken into consideration here are the video format conversion and Book Detection.

Use case 1: Video Conversion and Storage in Cloud Server.

The video file which is resource intensive is offloaded onto the server, because if the conversion is executed in smart phone it would take hours to get converted. Adding to that, the battery will be drained and storage space may not be enough for saving the converted video file whose size would have been increased. At client side, the user uploads the intensive resource to the cloud server. Once the user uploads the video conversion is done in the server and the user can download whenever necessary.
Use case 2: Book Detection

This is also an application that involves recognition of the name of the book where users will be intimated about the books’ reviews and ratings. So the users will have an opportunity to identify and save many dedicated information in cloud server.

At the end of application usage, the users are recommended to give ratings and review about the application. This would help when multiple user access the service.
Recommendation Engine based on Item-based Collaborative Filtering Algorithm

It is defined as depicted, “users having an identical outlook for similar things or items”. The similarity is based on how the individual users have rated it. It takes the base idea of the nearest-neighbor algorithm. The primary focus of item-based filtering is grounded on prediction. Rather than converging on users’ perception, item-item based similarities can be anticipated. The item defines the interest of the users which opted by them frequently, through which similar items will be recommended to the user. The similarity between different items can be designated in numerous ways; initially based on user ratings, secondly description about the product may also inspire, thirdly the co-occurrence of the item that is being frequently purchased and also based on previous history. The filtering is based on two factors.

Prediction: The prediction is based on the weighted average of rating and review given by the users.

Recommendation: Recommendations and suggestions given by several categories of people. It is calculated by the difference between the ratings or opinion given by the user for similar items over the average of the total users. Based on this it is recommended to other users.

The following pseudo code depicts the item-based CF procedure

```
For each application in the list A
  For each user U purchased the application A1
    For each application A2 purchased by the User U
      Include in record, the user purchase of A1 and A2
  For each application A2
    Compute similarity between A1 and A2
```

Cosine-based Similarity: The principle behind this is to calculate similarity of one item to the other. For

Figure 9 Sample rating by the user

Figure 8 Similarity Computation based on User Prediction
calculating similarity between two items, it can be manipulated by weighting the ratings of the users rated both. It is done by creating two vectors \((v_1, v_2)\) and finding cosine angle between them.

\[
\text{Sim}(i, j) = \cos(i, j) = \frac{(i \cdot j)}{\|i\| \cdot \|j\|}
\]

**Implementation of Item-based Model:**

<table>
<thead>
<tr>
<th>List of Users</th>
<th>Item 1</th>
<th>Item 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>User 1</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>User 2</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>User 3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>User 4</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>User 5</td>
<td>5</td>
<td>3</td>
</tr>
</tbody>
</table>

The prediction of results of application for \((u_1, u_2)\)

\[
\cos(u_1, u_2) = \frac{(4 \cdot 5 + 4 \cdot 4)}{\sqrt{(25 + 16) \cdot (16 \cdot 16)}} = 0.35
\]

Rating = number of users * cosine similarity

\[
\text{Rating} = (5 \cdot 0.35) = 1.75
\]

On collection of the tabulated reviews and ratings, predictive algorithm is used to calculate similarity index, with this similarity index prediction it can be recommended to other users.

**CONCLUSION**

This research work focuses on mobile-cloud infrastructure that is premeditated proficiently to manage the predictive analysis using collaborative-filtering algorithm. It also proposes the recommendation engine that filters user rating and review that helps other users to choose the service. This also enables context-aware client model using which remedies the user to make better use of service. This provides the user with improved platform that reduces and overhead and cost including high availability, reliability and scalability.

**REFERENCES**


15. A. Chen (2005), Context-aware collaborative ltering system: predicting the user’s preferences in ubiquitous computing, in *CHI EA*, 1110-1111.


