

A SITUATIONAL ANALYTIC METHOD FOR USER BEHAVIOR PATTERN IN MULTIMEDIA SOCIAL NETWORKS

VEERAVALLI ASHWINI¹, Dr.S.SATHPATHY²

¹M.Tech Student in CSE, NALANDA INSTITUTE OF ENGINEERING TECHNOLOGY, AP

²Professor, Dept of CSE, NALANDA INSTITUTE OF ENGINEERING TECHNOLOGY, AP

Abstract: The past decade has witnessed the emergence and progress of multimedia social networks (MSNs), which have explosively and tremendously increased to penetrate every corner of our lives, leisure and work. Moreover, mobile Internet and mobile terminals enable users to access to MSNs at anytime, anywhere, on behalf of any identity, including role and group. Therefore, the interaction behaviors between users and MSNs are becoming more comprehensive and complicated. This paper primarily extended and enriched the situation analytics framework for the specific social domain, named as *SocialSitu*, and further proposed a novel algorithm for users' intention serialization analysis based on classic Generalized Sequential Pattern (GSP). We leveraged the huge volume of user behaviors records to explore the frequent sequence mode that is necessary to predict user intention. Our experiment selected two general kinds of intentions: playing and sharing of multimedia, which are the most common in MSNs, based on the intention serialization algorithm under different minimum support threshold (*Min_Support*). By using the users' microscopic behaviors analysis on intentions, we found that the optimal behavior patterns of each user under the *Min_Support*, and a user's behavior patterns are different due to his/her identity variations in a large volume of sessions data.

I. INTRODUCTION

The rapid development of Multimedia Social Networks (MSNs) causes the tremendous growth of users and digital contents. It's also convenient for users to access digital contents in MSNs with a large-scale video dataset [1]. Meanwhile, the interaction between user and user, user and system increases. Therefore, providing users with timely and rapidly personalized services considering the complex interaction [2] is now a challenge in the study of multimedia social networks. Generally speaking, multimedia computing can be decomposed into three different stages, from data centric multimedia compression, content-centric multimedia communication and content analysis, to user-centric social media analysis till today, including user trust modeling [3, 4], propagation paths mining [5, 6] and digital right

sharing [7], and digital forensics[8-10]. However, understanding and predicting what multimedia content users' real needs in different situations and contexts have not been well studied [11]. Context-Aware (CA) [12-15] was first proposed by Schilit et al in 1994. They defined context as the set of location, people nearby, objects, and the changes of the objects. Prof. Carl K. Chang [16] proposed the Situ theory by combining the service environment with situation awareness to handle the dynamic update or development of service at run time. Therefore the service can meet the changing needs of users and provide users with personalized service. In order to adapt to the dynamic service environment and make a timely respond to the feedback of service environment, social media services increasingly require situation awareness. In social media networks, the human being is a complex and open system. The individual's intention can change at any time, which also causes a change in the user's needs. Moreover, the user's context and behavior are dynamic. Some studies show that the characteristics of the dynamic change will have different effects in a user's potential needs [17, 18]. A user's intention can be reflected through the acquiring attributes of the user's situation awareness and feedback on resources. The system can formulate a timely personalized service for the user based on user's intention, which will increase the user's service experience.

In social media networks, the user has different roles in different groups. The different identifications that the user has may cause the user's intention to change. The change of intention reflects the change in user's behavior. The Situ theory [16, 19] does not fully meet the analysis of the intention of users with different identities in the social media environment. This paper's motivation is to analyze the user's intention sequence mode(s) in social media networks. The major contributions of this paper are two folds. One is to enrich and extend the Situ theory outreaching for social domain, that is the social media ecosystem, through newly and comprehensively considering user's changeable identity (including role and group), and the other is to propose a novel algorithm for users' behavior pattern analysis and mining. The important vision of the work is to further predict users' more and deeper intention and mental based on a large volume of previous actions.

II. MODULE DESCRIPTION

Number of Modules:

After careful analysis the system has been identified to have the following:

Modules:

1. user Module
2. admin Module
3. chart module
4. video and image upload module

User module:

User module , the new user should register application form , before enter the particular site, after login , user should create the profile for that particular login user, then only unknown user or friends , can find out your profile based on your name or images or based on particular place, user can share images and unknown user can give comment for particular images, if unknown people give a any comment for your image, you can view that person profile , then you want to your friend list that particular person you can add it.

Upload and share Images:

User can share images and unknown user can give comment for particular images, if unknown people give a any comment for your image, you can view that person profile , then you want to your friend list that particular person you can add it. User can share any kind of image, friend list and other unknown user can view your images based on your privacy setting.

Comment for particular images:

User can give comment for share and upload images and give image related comment you can add that particular person in your friend list. Then you want to your friend list that particular person you can add it. User can share any kind of image, friend list and other unknown user can view your images based on your privacy setting.

Admin module:

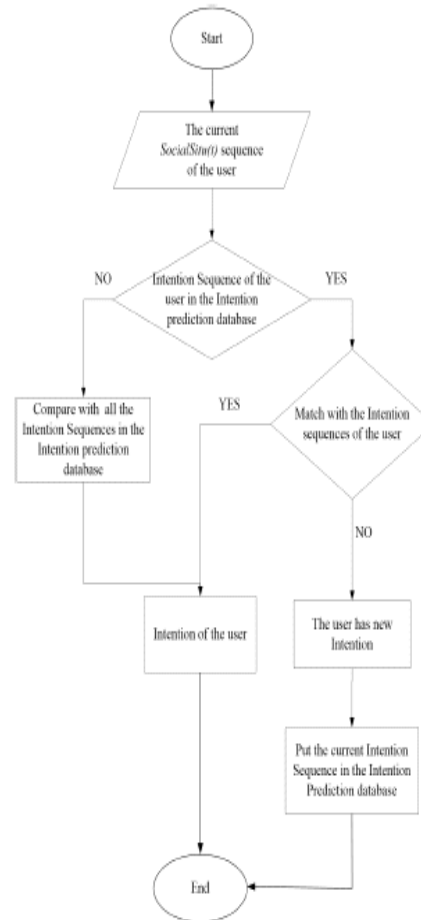
admin module, admin is a super user , admin can view the all the user details, admin can view the user prediction details also.. can view the chart based on user activities

Chart module:

The past decade has witnessed the emergence and progress of multimedia social networks (MSNs), which have explosively and tremendously increased to penetrate every corner of our lives, leisure and work. Moreover, mobile Internet and mobile terminals enable users to access to MSNs at anytime, anywhere, on behalf of

any identity, including role and group. Therefore, the interaction behaviors between users and MSNs are becoming more comprehensive and complicated.

architecture diagram:



III. SYSTEM TESTING

TESTING METHODOLOGIES

The following are the Testing Methodologies:

- **Unit Testing.**
- **Integration Testing.**
- **User Acceptance Testing.**
- **Output Testing.**
- **Validation Testing.**

Unit Testing

Unit testing focuses verification effort on the smallest unit of Software design that is the module. Unit testing exercises specific paths in a module's control structure to ensure complete coverage and maximum error detection. This test focuses on each module individually, ensuring that it functions properly as a unit. Hence, the naming is Unit Testing.

During this testing, each module is tested individually and the module interfaces are verified for the consistency with design specification. All important processing path are tested for the expected results. All error handling paths are also tested.

Integration Testing

Integration testing addresses the issues associated with the dual problems of verification and program construction. After the software has been integrated a set of high order tests are conducted. The main objective in this testing process is to take unit tested modules and builds a program structure that has been dictated by design.

The following are the types of Integration Testing:

1. Top Down Integration

This method is an incremental approach to the construction of program structure. Modules are integrated by moving downward through the control hierarchy, beginning with the main program module. The module subordinates to the main program module are incorporated into the structure in either a depth first or breadth first manner.

In this method, the software is tested from main module and individual stubs are replaced when the test proceeds downwards.

2. Bottom-up Integration

This method begins the construction and testing with the modules at the lowest level in the program structure. Since the modules are integrated from the bottom up, processing required for modules subordinate to a given level is always available and the need for stubs is eliminated. The bottom up integration strategy may be implemented with the following steps:

- The low-level modules are combined into clusters into clusters that perform a specific Software sub-function.
- A driver (i.e.) the control program for testing is written to coordinate test case input and output.
- The cluster is tested.
- Drivers are removed and clusters are combined moving upward in the program structure

The bottom up approaches tests each module individually and then each module is module is integrated with a main module and tested for functionality.

User Acceptance Testing

User Acceptance of a system is the key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with the prospective system users at the time of developing and making changes wherever required. The system developed provides a friendly user interface that can easily be understood even by a person who is new to the system.

Output Testing

After performing the validation testing, the next step is output testing of the proposed system, since no system could be useful if it does not produce the required output in the specified format. Asking the users about the format required by them tests the outputs generated or displayed by the system under consideration. Hence the output format is considered in 2 ways – one is on screen and another in printed format.

Validation Checking

Validation checks are performed on the following fields.

Text Field:

The text field can contain only the number of characters lesser than or equal to its size. The text fields are alphanumeric in some tables and alphabetic in other tables. Incorrect entry always flashes and error message.

Numeric Field:

The numeric field can contain only numbers from 0 to 9. An entry of any character flashes an error messages. The individual modules are checked for accuracy and what it has to perform. Each module is subjected to test run along with sample data. The individually tested modules are integrated into a single system. Testing involves executing the real data information is used in the program the existence of any program defect is inferred from the output. The testing should be planned so that all the requirements are individually tested.

A successful test is one that gives out the defects for the inappropriate data and produces and output revealing the errors in the system.

Preparation of Test Data

Taking various kinds of test data does the above testing. Preparation of test data plays a vital role in the system testing. After preparing the test data the system under study is tested using that test data. While testing the system by using test data

errors are again uncovered and corrected by using above testing steps and corrections are also noted for future use.

Using Live Test Data:

Live test data are those that are actually extracted from organization files. After a system is partially constructed, programmers or analysts often ask users to key in a set of data from their normal activities. Then, the systems person uses this data as a way to partially test the system. In other instances, programmers or analysts extract a set of live data from the files and have them entered themselves.

It is difficult to obtain live data in sufficient amounts to conduct extensive testing. And, although it is realistic data that will show how the system will perform for the typical processing requirement, assuming that the live data entered are in fact typical, such data generally will not test all combinations or formats that can enter the system. This bias toward typical values then does not provide a true systems test and in fact ignores the cases most likely to cause system failure.

Using Artificial Test Data:

Artificial test data are created solely for test purposes, since they can be generated to test all combinations of formats and values. In other words, the artificial data, which can quickly be prepared by a data generating utility program in the information systems department, make possible the testing of all login and control paths through the program.

The most effective test programs use artificial test data generated by persons other than those who wrote the programs. Often, an independent team of testers formulates a testing plan, using the systems specifications.

The package "Virtual Private Network" has satisfied all the requirements specified as per software requirement specification and was accepted.

USER TRAINING

Whenever a new system is developed, user training is required to educate them about the working of the system so that it can be put to efficient use by those for whom the system has been primarily designed. For this purpose the normal working of the project was demonstrated to the prospective users. Its working is easily understandable and since the expected users are

people who have good knowledge of computers, the use of this system is very easy.

MAINTAINENCE

This covers a wide range of activities including correcting code and design errors. To reduce the need for maintenance in the long run, we have more accurately defined the user's requirements during the process of system development. Depending on the requirements, this system has been developed to satisfy the needs to the largest possible extent. With development in technology, it may be possible to add many more features based on the requirements in future. The coding and designing is simple and easy to understand which will make maintenance easier.

TESTING STRATEGY :

A strategy for system testing integrates system test cases and design techniques into a well planned series of steps that results in the successful construction of software. The testing strategy must co-operate test planning, test case design, test execution, and the resultant data collection and evaluation .A strategy for software testing must accommodate low-level tests that are necessary to verify that a small source code segment has been correctly implemented as well as high level tests that validate major system functions against user requirements.

Software testing is a critical element of software quality assurance and represents the ultimate review of specification design and coding. Testing represents an interesting anomaly for the software. Thus, a series of testing are performed for the proposed system before the system is ready for user acceptance testing.

SYSTEM TESTING:

Software once validated must be combined with other system elements (e.g. Hardware, people, database). System testing verifies that all the elements are proper and that overall system function performance is achieved. It also tests to find discrepancies between the system and its original objective, current specifications and system documentation.

UNIT TESTING:

In unit testing different are modules are tested against the specifications produced during the design for the modules. Unit testing is essential for verification of the code produced during the coding phase, and hence the goals to test the internal logic of the modules. Using the detailed design description as a guide, important Conrail

paths are tested to uncover errors within the boundary of the modules. This testing is carried out during the programming stage itself. In this type of testing step, each module was found to be working satisfactorily as regards to the expected output from the module.

In Due Course, latest technology advancements will be taken into consideration. As part of technical build-up many components of the networking system will be generic in nature so that future projects can either use or interact with this. The future holds a lot to offer to the development and refinement of this project.

IV. CONCLUSION

The existing MSNs environment increasingly requires situation awareness. Users' environment and behavior are dynamic, and an individual's intention is also to change. In order to adapt to the dynamic changes of user identities in the social domain, this paper extends and enriches the Situ theory, and builds a *SocialSitu* framework for the social media networks. We design and achieve the intention serialization algorithm in multimedia social networks. The user's frequent intention sequence mode is obtained through the intention serialization algorithm. When the user's identify changes, we conclude his behavior pattern with different ID, and prove that different *SocialSitu(t)* sequences are acquired in the same *Min_Support* with the same intention when his role and group change. In the future works, the existing intention sequence patterns of the user could be adopted to predict the user's more and deeper intentions. Besides, we will employ the *SocialSitu* and the proposed algorithm to improve multimedia recommendation system and some killer applications in MSNs

Future enhancement:

In future, we will future develop our algorithm in the following aspects: In the future works, the existing intention sequence patterns of the user could be adopted to predict the user's more and deeper intentions. Besides, we will employ the *SocialSitu* and the proposed algorithm to improve multimedia recommendation system and some killer applications in MSNs

REFERENCES

- [1] Y. G. Jiang and J. J. Wang, "Partial Copy Detection in Videos: A Benchmark and an Evaluation of Popular Methods," IEEE Trans. Big Data, vol. 2, no. 1, pp. 32-42, Jan/Mar 2016, doi:10.1109/TBDATA.2016.2530714.
- [2] B. De Meester, R. Verborgh, P. Pauwels, W. De Neve, E. Mannens, and R. Van de Walle, "Towards robust and reliable multimedia analysis through semantic integration of services," Multimedia Tools Appl., vol. 75, no. 22, pp. 14019-14038, Nov. 2016.
- [3] Z. Zhang and K. Wang, "A Trust Model for Multimedia Social Networks," Soc. Netw. Anal. Min., vol. 3, no. 4, pp. 969-979, Dec. 2013.
- [4] Z. Zhang and B. B. Gupta, "Social Media Trustworthiness and Security: Overview and New Direction," Future Generation Computer Systems, submitted for publication.
- [5] W. Feng, Z. Zhang, J. Wang, and L. Han, "A Novel Authorization Delegation for Multimedia Social Networks by using Proxy Re-encryption," Multimedia Tools Appl., vol. 75, no. 21, pp. 13995-14014, Nov. 2016.
- [6] Z. Zhang and K. Wang, "A Formal Analytic Approach to Credible Potential Path and Mining Algorithms for Multimedia Social Networks," Comput J., vol. 58, no.4, pp. 668- 678, Sep. 2015.
- [7] Z. Zhang, Z. Wang, and D. Niu, "A Novel Approach to Rights Sharing-Enabling Digital Rights Management for Mobile Multimedia," Multimedia Tools Appl., vol. 74, no. 16, pp. 6255-6271, Aug. 2015.
- [8] A. Azfar, K.-K. R. Choo, and L. Liu, "Forensic Taxonomy of Android Social Apps," J. Forensic Sci., preprint, Nov. 2016, doi: 10.1111/1556-4029.13267.
- [9] D. Quick and K.-K. R. Choo, "Big forensic data management in heterogeneous distributed systems: quick analysis of multimedia forensic data," Softw. Pract. Exper., preprint, 2016, doi: 10.1002/spe.2429.
- [10] N. H. Ab Rahman, W. B. Glisson, Y. Yang, and K.-K. R. Choo, "Forensic-by-design framework for cyber-physical cloud systems," IEEE Cloud Comput., vol. 3, no. 1, pp. 50-59, Feb 2016.
- [11] P. Cui, W. Zhu, T. S. Chua, and R. Jain, "Social-Sensed Multimedia Computing," IEEE Multimedia, vol. 23, no. 1, pp. 92-96, Jan/Mar 2016.
- [12] B. N. Schilit and M. M. Theimer, "Disseminating Active Map Information to Mobile Hosts," IEEE Network, vol. 8, no. 5, pp. 22-32, Sep/Oct 1994.
- [13] K. Srinivasan, P. Agrawal, R. Arya, N. Akhtar, D. Pengoria, and T. A. Gonsalves, "Context-aware, QoE-driven adaptation of multimedia services," 5th International Conference on Mobile Wireless Middleware, Operating Systems, and Applications, pp. 236-249, Nov. 2012, doi: 10.1007/978-3-642-36660-4_17.
- [14] C. Tekin and M. Van Der Schaar, "Contextual online learning for multimedia content aggregation," IEEE Trans. Multimedia, vol. 17, no. 4, pp. 549-561, Apr. 2015, doi: 10.1109/TMM.2015.2403234.