International Journal of Research
ISSN: 2348-6848 Vol-3, Special Issue-1
National Conference on Advanced Computing Technologies
Held on 21st July 2015 at Hyderabad city organized by Global Research Academy, Hyderabad, Telangana, India.

Design and Development of a Location–Aware Recommender System for Producing More Accurate and Reliable Recommendations to Maximize System Scalability

B.Sindhuja[1]; Sai Manogna[2]&Prof.Dr.G.Manoj Someswar[3]

1. **M.Tech.(CSE) from Narasimha Reddy Engineering College, Affiliated to JNTUH, Hyderabad, Telangana, India**
2. **M.Tech. (CSE), Assistant Professor, Department of CSE, Narasimha Reddy Engineering College, Affiliated to JNTUH, Hyderabad, Telangana, India**
3. **B.Tech., M.S.(USA), M.C.A., Ph.D., Principal & Professor, Department Of CSE, Anwar-ul-uloom College of Engineering & Technology, Affiliated to JNTUH, Vikarabad, Telangana, India**

ABSTRACT: *This paper proposes LARS*, a location-aware recommender system that uses location-based ratings to produce recommendations. Traditional recommender systems do not consider spatial properties of users nor items; LARS*, on the other hand, supports a taxonomy of three novel classes of location-based ratings, namely, spatial ratings for non-spatial items, non-spatial ratings for spatial items, and spatial ratings for spatial items. LARS* exploits user rating locations through user partitioning, a technique that influences recommendations with ratings spatially close to querying users in a manner that maximizes system scalability while not sacrificing recommendation quality. LARS* exploits item locations using travel penalty, a technique that favors recommendation candidates closer in travel distance to querying users in a way that avoids exhaustive access to all spatial items. LARS* can apply these techniques separately, or together, depending on the type of location-based rating available. Experimental evidence using large-scale real-world data from both the Foursquare location-based social network and the MovieLens movie recommendation system reveals that LARS* is efficient, scalable, and capable of producing recommendations twice as accurate compared to existing recommendation approaches.*

Keywords: Location Aware Recommender System; Classification and Regression Trees (CART); Chi Square Automatic Interaction Detection (CHAID); Non-Spatial Items; Spatial Ratings; Knowledge Discovery

INTRODUCTION

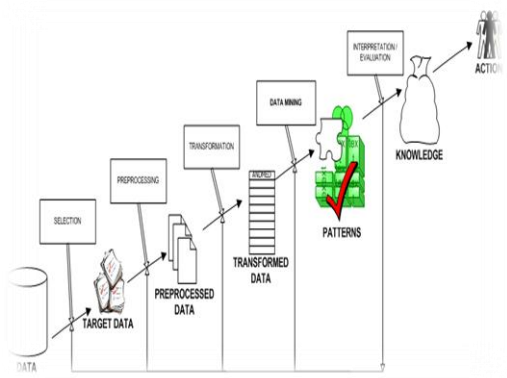

Figure 1: Structure of Data Mining

Generally, data mining (sometimes called data or knowledge discovery) is the process of analyzing data from different perspectives and summarizing it into useful information - information that can be used to increase revenue, cuts costs, or both. Data mining software is one of a number of analytical tools for analyzing data. It allows users to analyze data from many different dimensions or angles, categorize it, and summarize the relationships identified. Technically, data mining is the process of finding correlations or patterns among dozens of fields in large relational databases.[1]

While large-scale information technology has been evolving separate transaction and analytical systems, data mining provides the link between the two. Data mining software analyzes relationships and patterns in stored transaction data based on open-ended user queries. Several types of analytical software are available: statistical, machine learning, and neural networks.[2]**Generally, any of four types of relationships are sought:**

- **Classes**: Stored data is used to locate data in predetermined groups. For example, a restaurant chain could mine customer purchase data to determine when customers visit and what they typically order. This information could be used to increase traffic by having daily specials.
- **Clusters**: Data items are grouped according to logical relationships or consumer preferences. For example, data can be mined to identify market segments or consumer affinities.
- **Associations:** Data can be mined to identify associations. The beer-diaper example is an example of associative mining.
- **Sequential patterns**: Data is mined to anticipate behavior patterns and trends. For example, an outdoor equipment retailer could predict the likelihood of a backpack being purchased based on a consumer's purchase of sleeping bags and hiking shoes.[3]

Data mining consists of five major elements:

Extract, transform, and load transaction data onto the data warehouse system.

1) Store and manage the data in a multidimensional database system.
2) Provide data access to business analysts and information technology professionals.
3) Analyze the data by application software.
4) Present the data in a useful format, such as a graph or table.[4]

Different levels of analysis are available:

- **Artificial neural networks:** Non-linear predictive models that learn through training and resemble biological neural networks in structure.
- **Genetic algorithms**: Optimization techniques that use process such as genetic combination, mutation, and natural selection in a design based on the concepts of natural evolution.
- **Decision trees**: Tree-shaped structures that represent sets of decisions. These decisions generate rules for the classification of a dataset. Specific decision tree methods include Classification and Regression Trees (CART) and Chi Square Automatic Interaction Detection (CHAID). CART and CHAID are decision tree techniques used for classification of a dataset. They provide a set of rules that you can apply to a new (unclassified) dataset to predict which records will have a given outcome. CART segments a dataset by creating 2-way splits while CHAID segments using chi square tests to create multi-way splits. CART typically requires less data preparation than CHAID.
- **Nearest neighbor method**:
  A technique that classifies each record in a dataset based on a combination of the classes of the *k* record(s) most similar to it in a historical dataset (where *k*=1). Sometimes called the *k*-nearest neighbor technique.
- **Rule induction**: The extraction of useful if-then rules from data based on statistical significance.
- **Data visualization**: The visual interpretation of complex relationships in multidimensional data. Graphics tools are used to illustrate data relationships.[5]

Characteristics of Data Mining:

- **Large quantities of data**: The volume of data so great it has to be analyzed by automated techniques e.g. satellite information, credit card transactions etc.
- **Noisy, incomplete data**: Imprecise data is the characteristic of all data collection.

- **Complex data structure**: conventional statistical analysis not possible
- **Heterogeneous data stored in legacy systems[6]**

Benefits of Data Mining:

1) It's one of the most effective services that are available today. With the help of data mining, one can discover precious information about the customers and their behavior for a specific set of products and evaluate and analyze, store, mine and load data related to them
2) An analytical CRM model and strategic business related decisions can be made with the help of data mining as it helps in providing a complete synopsis of customers
3) An endless number of organizations have installed data mining projects and it has helped them see their own companies make an unprecedented improvement in their marketing strategies (Campaigns)
4) Data mining is generally used by organizations with a solid customer focus. For its flexible nature as far as applicability is concerned is being used vehemently in applications to foresee crucial data including industry analysis and consumer buying behaviors
5) Fast paced and prompt access to data along with economic processing techniques have made data mining one of the most suitable services that a company seek.[7]

Advantages of Data Mining:

1. Marketing / Retail:

Data mining helps marketing companies build models based on historical data to predict who will respond to the new marketing campaigns such as direct mail, online marketing campaign…etc. Through the results, marketers will have appropriate approach to sell profitable products to targeted customers.

Data mining brings a lot of benefits to retail companies in the same way as marketing. Through market basket analysis, a store can have an appropriate production arrangement in a way that customers can buy frequent buying products together with pleasant. In addition, it also helps the retail companies offer certain discounts for particular products that will attract more customers.

2. Finance / Banking

Data mining gives financial institutions information about loan information and credit reporting. By building a model from historical customer's data, the bank and financial institution can determine good and bad loans. In addition, data mining helps banks detect fraudulent credit card transactions to protect credit card's owner.

3. Manufacturing

By applying data mining in operational engineering data, manufacturers can detect faulty equipments and determine optimal control parameters. For example semi-conductor manufacturers has a challenge that even the conditions of manufacturing environments at different wafer production plants are similar, the quality of wafer are lot the same and some for unknown reasons even has defects. Data mining has been applying to determine the ranges of control parameters that lead to the production of golden wafer. Then those optimal control parameters are used to manufacture wafers with desired quality.

4. Governments

Data mining helps government agency by digging and analyzing records of financial transaction to build patterns that can detect money laundering or criminal activities.

5. Law enforcement:

Data mining can aid law enforcers in identifying criminal suspects as well as apprehending these criminals by examining trends in location, crime type, habit, and other patterns of behaviors.

6. Researchers:

Data mining can assist researchers by speeding up their data analyzing process; thus, allowing those more time to work on other projects. [8]

LITERATURE SURVEY

This paper presents an overview of the field of recommender systems and describes the current generation of recommendation methods that are usually classified into the following three main categories: content-based, collaborative, and hybrid recommendation approaches. This paper also describes various limitations of current recommendation methods and discusses possible extensions that can improve recommendation capabilities and make recommender systems applicable to an even broader range of applications. These extensions include, among others, an improvement of understanding of users and items, incorporation of the contextual information into the recommendation process, support for multcriteria ratings, and a provision of more flexible and less intrusive types of recommendations.[9]

This paper proposes LARS, a location-aware recommender system that uses location-based ratings to produce recommendations. Traditional recommender systems do not consider spatial properties of users nor items, LARS, on the other hand, supports taxonomy of three novel classes of location-based ratings, namely, spatial ratings for non-spatial items, non-spatial ratings for spatial items, and spatial ratings for spatial items. LARS exploits user rating locations through user partitioning, a technique that influences recommendations with ratings spatially close to querying users in a manner that maximizes system scalability while not sacrificing recommendation quality. LARS exploits item locations using travel penalty, a technique that favors recommendation candidates closer in travel distance to querying users in a way that avoids exhaustive access to all spatial items. LARS can apply these techniques separately, or in concert, depending on the type of location-based rating available. Experimental evidence using large-scale real-world data from both the foursquare location-based social network and the Movie Lens movie recommendation system reveals that LARS is efficient, scalable, and capable of producing recommendations twice as accurate compared to existing recommendation approaches.[10] Collaborative filtering or recommender systems use a database about user preferences to predict additional topics or products a new user might like. In this paper we describe several algorithms designed for this task, including techniques based on correlation coefficients, vector-based similarity calculations, and statistical Bayesian methods. We compare the predictive accuracy of the various methods in a set of representative problem domains. We use two basic classes of evaluation metrics. The first characterizes accuracy over a set of individual predictions in terms of average absolute deviation. The second estimates the utility of a ranked list of suggested items. This metric uses an estimate of the probability that a user will see a recommendation in an ordered list. Experiments were run for datasets associated with 3 application areas, 4 experimental protocols, and the 2 evaluation metrics for the various algorithms. Results indicate that for a wide range of conditions, Bayesian networks with decision trees at each node and correlation methods outperform Bayesian-clustering and vector-similarity methods. Between correlation and Bayesian networks, the preferred method depends on the nature of the dataset, nature of the application (ranked versus one-by-one presentation), and the availability of votes with which to make predictions. Other considerations include the size of database, speed of predictions, and learning time.

Window operations serve as the basis of a number of queries that can be posed in a spatial database. Examples of these window-based queries include the exist query (i.e., determining whether or not a spatial feature exists inside a window) and the report query, (i.e., reporting the identity of all the features that exist inside a window).[11] Algorithms are described for answering window queries in &Ogr;(n log logT) time for a window of size n x n in a feature space (e.g., an image) of size T x T (e.g., pixel elements). The significance of this result is that even though the window contains n2 pixel elements, the worst-case

time complexity of the algorithms is almost linearly proportional (and not quadratic) to the window diameter, and does not depend on other factors. The above complexity bounds are achieved via the introduction of the incomplete pyramid data structure (a variant of the pyramid data structure) as the underlying representation to store spatial features and to answer queries on them.[12]

Recommender systems have been evaluated in many, often incomparable, ways. In this article, we review the key decisions in evaluating collaborative filtering recommender systems:[13] the user tasks being evaluated, the types of analysis and datasets being used, the ways in which prediction quality is measured, the evaluation of prediction attributes other than quality, and the user-based evaluation of the system as a whole.[14] In addition to reviewing the evaluation strategies used by prior researchers, we present empirical results from the analysis of various accuracy metrics on one content domain where all the tested metrics collapsed roughly into three equivalence classes. Metrics within each equivalency class were strongly correlated, while metrics from different equivalency classes were uncorrelated.[15]

SYSTEM STUDY

FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- ECONOMICAL FEASIBILITY
- TECHNICAL FEASIBILITY
- SOCIAL FEASIBILITY

ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

SYSTEM DESIGN

SYSTEM ARCHITECTURE:

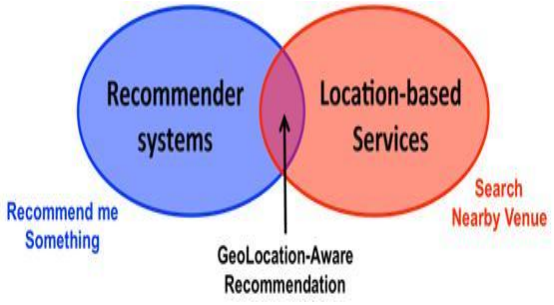

Figure 2: System Architecture

DATA FLOW DIAGRAM:

1. The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.
2. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.
3. DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.
4. DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.

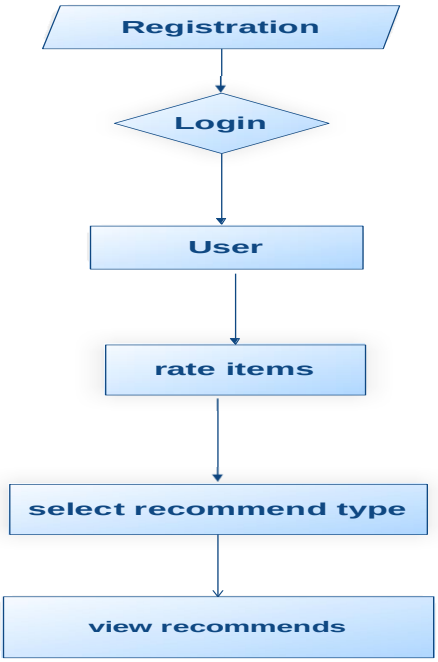

Figure 3: Data Flow Diagram

UML DIAGRAMS

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing objects oriented software and the software

development process. The UML uses mostly graphical notations to express the design of software projects.

GOALS:

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

USE CASE DIAGRAM:

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

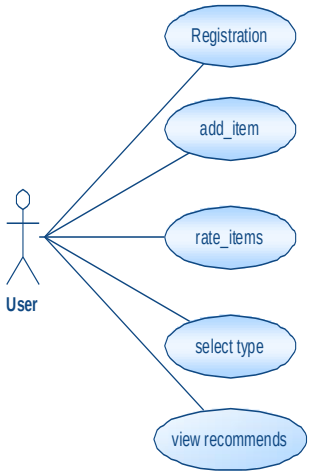

Figure 4: Use Case Diagram

CLASS DIAGRAM:

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

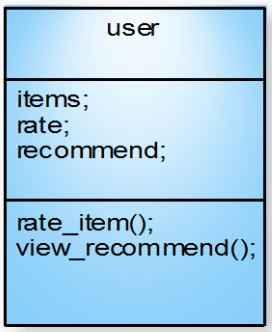

Figure 5: Class Diagram

SEQUENCE DIAGRAM:

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

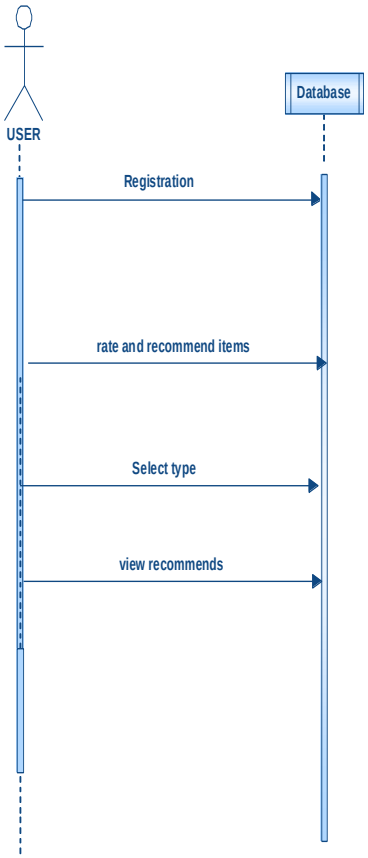

Figure 6: Sequence Diagram

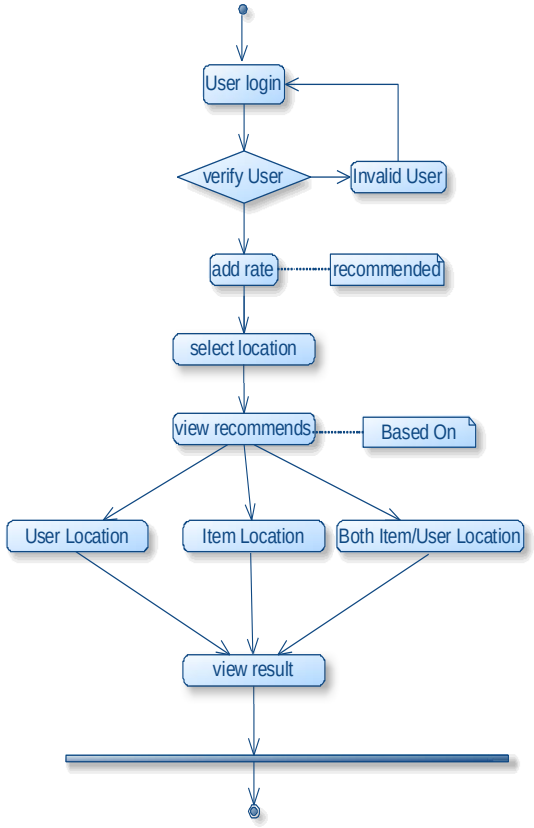

Figure 7: Activity Diagram

ACTIVITY DIAGRAM:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

INPUT DESIGN

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

- What data should be given as input?
- How the data should be arranged or coded?
- The dialog to guide the operating personnel in providing input.

- Methods for preparing input validations and steps to follow when error occur.

OBJECTIVES

1.Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.

2.It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.

3.When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user

will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow

OUTPUT DESIGN

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

1. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.

2.Select methods for presenting information.

3.Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

- Convey information about past activities, current status or projections of the
- Future.
- Signal important events, opportunities, problems, or warnings.
- Trigger an action.
- Confirm an action.

SYSTEM ANALYSIS

EXISTING SYSTEM:

Recommender systems make use of community opinions to help users identify useful items from a considerably large search space. The technique used by many of these systems is collaborative filtering (CF), which analyzes past community opinions to find correlations of similar users and items to suggest k personalized items (e.g., movies) to a querying user u. Community opinions are expressed through explicit ratings represented by the triple (user, rating, item) that represents a user providing a numeric rating for an item. Myriad applications can produce location-based ratings that embed user and/or item locations. Existing recommendation techniques assume ratings are represented by the (user, rating, item) triple.

DISADVANTAGES OF EXISTING SYSTEM:

- The existing systems are ill-equipped to produce location aware recommendations.
- The existing system provides more expensive operations to maintain the user partitioning structure.
- The existing system does not provide spatial ratings.

PROPOSED SYSTEM:

We have proposed LARS*, a location-aware recommender system that uses location-based ratings to produce recommendations. LARS*, supports a taxonomy of three novel classes of location-based ratings, namely, spatial ratings for non-spatial items,

non-spatial ratings for spatial items, and spatial ratings for spatial items. LARS* exploits user rating locations through user partitioning, a technique that influences recommendations with ratings spatially close to querying users in a manner that maximizes system scalability while not sacrificing recommendation quality. LARS* exploits item locations using travel penalty, a technique that favors recommendation candidates closer in travel distance to querying users in a way that avoids exhaustive access to all spatial items. LARS* can apply these techniques separately, or together, depending on the type of location-based rating available. Within LARS*, we propose:

(a) A user partitioning technique that exploits user locations in a way that maximizes system scalability while not sacrificing recommendation locality

(b) A travel penalty technique that exploits item locations and avoids exhaustively processing all spatial recommendation candidates.

ADVANTAGES OF PROPOSED SYSTEM:

- LARS*, supports a taxonomy of three novel classes of location-based ratings, namely, spatial ratings for non-spatial items, non-spatial ratings for spatial items, and spatial ratings for spatial items.
- LARS* achieves higher locality gain using a better user partitioning data structure and algorithm.
- LARS* exhibits a more flexible tradeoff between locality and scalability.
- LARS* provides a more efficient way to maintain the user partitioning structure

SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that theSoftware system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

TYPES OF TESTS

Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.
Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot "see" into it. The test provides inputs and responds to outputs without considering how the software works.

Unit Testing:

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

IMPLEMENTATION

MODULE:

1. Spatial ratings for non-spatial items
2. Non-spatial ratings for spatial items
3. Spatial ratings for spatial items

MODULES DESCRIPTION:

Spatial ratings for non-spatial items:

This section describes how LARS* produces recommendations using spatial ratings for non-spatial items represented by the tuple (user, location, rating, item). The idea is to exploit preference locality, i.e., the observation that user opinions are spatially unique. We identify three requirements for producing recommendations using spatial ratings for non-spatial items: (1) Locality: recommendations should be influenced by those ratings with user locations spatially close to the querying user location (i.e., in a spatial neighborhood); (2) Scalability: the recommendation procedure and data structure should scale up to large number of users; (3) Influence: system users should have the ability to control the size of the spatial neighborhood (e.g., city block, zip code, or county) that influences their recommendations.

Non-spatial ratings for spatial items:

This section describes how LARS* produces recommendations using non-spatial ratings for spatial items represented by the tuple (user, rating, item, location). The idea is to exploit travel locality, i.e., the observation that users limit their choice of spatial venues based on travel distance. Traditional (non-spatial) recommendation techniques may produce recommendations with burdensome travel distances (e.g., hundreds of miles away). LARS* produces recommendations within reasonable travel distances by using travel penalty, a technique that penalizes the recommendation rank of items the further in travel distance they are from a querying user. Travel penalty may incur expensive computational overhead by calculating travel distance to each item. Thus, LARS* employs an efficient query processing technique capable of early termination to produce the recommendations with-out calculating the travel distance to all items.

Spatial ratings for spatial items:

This section describes how LARS* produces recommendations using spatial ratings for spatial items represented by the tuple (user, location, rating, item, location). A salient feature of LARS* is that both the user partitioning and travel

penalty techniques can be used together with very little change to produce recommendations using spatial user ratings for spatial items. The data structures and maintenance techniques remain exactly the same as discussed in Sections 4and5; only the query processing frame-work requires a slight modification. Query processing uses Algorithm2 to produce recommendations. However, the only difference is that the item-based collaborative filtering prediction score P(u,i) used in the recommendation score calculation (Line16in Algorithm 2) is generated using the (localized) collaborative filtering model from the partial pyramid cell that contains the querying user, instead of the system-wide collaborative filtering model as was used.

RESULTS &CONCLUSION

LARS*, our proposed location-aware recommender system, tackles a problem untouched by traditional recommender systems by dealing with three types of location-based ratings: *spatial ratings for non-spatial items*, *non-spatial ratings for spatial items*, and *spatial ratings for spatial items*. LARS* employs *user partitioning* and *travel penalty* techniques to support spatial ratings and spatial items, respectively. Both techniques can be applied separately or in concert to support the various types of location-based ratings. Experimental analysis using real and synthetic data sets show that LARS* is efficient, scalable, and provides better quality recommendations than techniques used in traditional recommender systems.

REFERENCES

[1] G. Linden, B. Smith, and J. York, "Amazon.com recommendations: Item-to-item collaborative filtering," *IEEE Internet Comput.*, vol. 7, no. 1, pp. 76–80, Jan./Feb. 2003.

[2] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "GroupLens: An open architecture for

collaborative filtering of netnews," in *Proc. CSWC*, Chapel Hill, NC, USA, 1994.

[3] The facebook blog. *Facebook Places* [Online]. Available: http://tinyurl.com/3aetfs3

[4] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 6, pp. 734–749, Jun. 2005.

[5] *MovieLens* [Online]. Available: http://www.movielens.org/

[6] *Foursquare* [Online]. Available: http://foursquare.com

[7] *New York Times - A Peek into Netflix Queues* [Online]. Available: http://www.nytimes.com/interactive/2010/01/10/nyregion/20100110-netflix-map.html

[8] J. J. Levandoski, M. Sarwat, A. Eldawy, and M. F. Mokbel, "LARS: A location-aware recommender system," in *Proc. ICDE*, Washington, DC, USA, 2012.

[9] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proc. Int. Conf. WWW*, Hong Kong, China, 2001.

[10] J. S. Breese, D. Heckerman, and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," in *Proc. Conf. UAI*, San Francisco, CA, USA, 1998.

[11] W. G. Aref and H. Samet, "Efficient processing of window queries in the pyramid data structure," in *Proc. ACM Symp. PODS*, New York, NY, USA, 1990.

[12] R. A. Finkel and J. L. Bentley, "Quad trees: A data structure for retrieval on composite keys," *Acta Inf.*, vol. 4, no. 1, pp. 1–9, 1974.

[13] A. Guttman, "R-trees: A dynamic index structure for spatial searching," in *Proc. SIGMOD*, New York, NY, USA, 1984.

[14] K. Mouratidis, S. Bakiras, and D. Papadias, "Continuous monitoring of spatial queries in wireless broadcast environments," *IEEE Trans. Mobile Comput.*, vol. 8, no. 10, pp. 1297–1311, Oct. 2009.

[15] K. Mouratidis and D. Papadias, "Continuous nearest neighbor queries over sliding windows," *IEEE Trans. Knowl. Data Eng.*, vol. 19, no. 6, pp. 789–803, Jun. 2007.