CHAPTER 10

COMPUTER SCIENCE

Doctoral Theses

01. CHAUDHARY (Reshu)
Enhancing Efficiency in Swarm Intelligence Algorithms towards Novel Peacock Algorithm.
Supervisor : Dr. Hema Banati
Th 25526

Abstract
(Not Verified)

Swarm intelligence (SI) algorithms are nature inspired optimization techniques. The efficiency of these algorithms is based on their ability to balance between exploration and exploitation. The aim of this research is to explore various factors like range of random movement, population partitioning, diversity, parameter setting, etc which influence this balance. The research work involved studying multiple existing techniques for these factors, and developing new techniques wherever needed. Different techniques for enhancing the exploration capabilities, like weighted multi-modal technique, numerous population partitioning techniques, diversity based replacement techniques, employing different parameters, etc. were developed. These techniques efficiently enhanced overall diversity of SI algorithms, thus imparting them a better chance of escaping local optima. An improved search technique for exploitation capabilities was developed which proved to be efficient in improving the overall efficiency of multiple SI algorithms. Different techniques which focus on exploration and exploitation simultaneously were also developed, like swarming with improved search, adaptive multi-population communication technique, weighted multi-modal technique with improved search, etc. Based on the knowledge gained by this study, a novel SI algorithm, the peacock algorithm (PA) was developed. PA is based on the lekking behaviour of peacocks. A lek is an aggregation of peacocks, where every peacock defends its own mating/lekking range. The best solutions are identified as peacocks, while the remaining solutions are called peahens. Peahens are assigned to peacocks based on their relative positions. Peahens move towards their respective lek peacocks, while peacocks move towards the global best peacock. The global best peacock maintains his lek by pushing away other peacocks if they encroach upon its lekking area. PA is compared to nine SI algorithms, and results establish its exploration and exploitation efficiency. The findings of this research provide guidelines that will benefit the study of understanding and enhancing swarm intelligence algorithm in a wider context.

Contents

02. CHAUHAN (Jyoti)
Framework for Learning Components in MOOC Platform.
Supervisor : Dr. Anita Goel
Th 24988

Abstract
(Not Verified)

Massive Open Online Course (MOOC) provides the online courses that are open for massive participation and are accessible to everyone via the web. In MOOC platform, the learning is characterised by mainly three components, namely, video, quiz, and, social networking and collaboration, which are integrated in the platform. For integrating the learning components in a MOOC platform, availability of specification documents is required for different phases of software development life cycle. Since the specification documents of learning components are not available, there are some issues involved during their integration in MOOC. The video, quiz, and discussion forum from collaboration, are integrated in MOOC platform as software. For their integration in MOOC platform, either already existing software is used or new software is developed. But there is no requirement specification document available for the same. It forces the repeated software elicitation for new software developed. Also, when using existing software it is not easy to identify their features. Moreover, there is a need to prioritize the requirements, since not all requirements are equally important. Additionally, there is an absence of design documents for learning components, making the updating of software a cumbersome task. Moreover, social networking is integrated as functionality directly in the MOOC platform. Collaboration tools are incorporated in the platform. So their issues need to be handled separately. Therefore, there is a need for the specification documents for each learning component to be used in software development, to ease their integration in a MOOC platform. In this thesis, we focus on software specification documents required during integration of learning components in a MOOC platform - (1) requirement document for learning components, (2) software requirement prioritization and (3) software design document. We have developed a generic framework to be used during the integration of learning components in a MOOC platform.

Contents

03. GUPTA (Shalini)
Intelligent Recommender Systems Based on Clickstream Sessions.
Supervisor : Dr. V. S. Dixit
Th 24992

Abstract
(Verified)

Recommender System is a tool that provides personalized recommendations to customers and helps them in decision making process. This research aims to explore novel approaches for recommending products that uses implicit behavior of customers. Clickstream data is one such form of implicit data in which user's navigational and behavioral parameters are traced to recommend most suitable products. Various classifiers are used to find the preference of products.. For the products that are not clicked by the target user, like-minded users are grouped to predict the preference level. After calculating the respective preference levels of all the categorized products, sequential behavior of a customer is taken into account. A

hybrid framework is developed to deploy both the techniques and top-N recommendations are generated. Also, customers are clustered on the basis of various context clicks. Preference of individual user is predicted for each context and is stored in user-context preference matrix. Similarly user-item category matrix is built. A recommender algorithm is also designed using Association Rule Mining technique. Here the products that are clicked by similar users are marked to find association level among the products using association rule mining to generate user-product preference matrix. The attributes identified from clickstream sessions reflects absolute preference of a customer towards a product viewed. The preferences are further refined by calculating the sequential preference of user for the products. The algorithm proposes an intelligent recommender system known as SAPRS (Sequential Absolute Preference based Recommender System) that makes use of these two approaches which are integrated to improve the quality of recommendation. Therefore, this research will provide a solution to ecommerce websites to maintain their customers using clickstream sessions. The techniques employing implicit behavior of customers outperform state of the art methods in terms of precision, recall and F1 measures.

Contents

04. JAIN (Parul)
Improving Quality of Recommendations for Context Aware Collaborative Filtering.
Supervisor : Dr. V. S. Dixit
Th24991

Abstract
(Not Verified)

Context aware recommender system which has become a hot research topic because it carries a strong intuitive appeal. It endorses that preferences of users highly depend on situation. The proposed work in this thesis aims at providing more relevant recommendations through context aware collaborative filtering techniques addressing various challenges. In this thesis, contextual information is applied to collaborative filtering approach in three different ways. In the first methodology, context features statistically found valid and influential in the previous studies are used. The algorithms in second phase focus on finding and utilizing similarity in contextual situation of the users. Also, some novel similarity measures are introduced that are used by collaborative filtering algorithms. The third part demonstrates algorithms implementing optimal contribution of all context features. Particle swarm optimization has been used to find optimal weight of each context feature. The objective of using PSO is to include all context features with their appropriate weight. Further, clustering is used in neighborhood formation to improve the scalability and accuracy of the algorithm which is a new emerging topic of research nowadays. Clustering helps in formation of better neighborhood since most similar preferences cluster together by these techniques. The performance of all proposed techniques are evaluated under different scenarios for individuals and even in some cases for group of users. The presented algorithms are also compared with existing techniques in the literature. Moreover, the results obtained show that the context aware collaborative filtering techniques proposed in this work are able to produce better quality recommendations compared to existing

approaches. These techniques are applicable in any real life recommendation problem independent of the number and type of context dimensions.

Contents

1. Introduction 2. Background 3. Eliciting contextual preference and computing 4. Impact of expect/ implicit ratings and weighted percentile approach 5. Learning similarity measures using context and other measures 6. Improved similarity measures for enhanced context aware recommendations 7. Sparsity based fusion of local and global similarity 8. Quality recommendations based on clustering and weighted context 9. Conclusion. List of publications. References.

05. MADHU KUMARI
Entropy Based Approach for Software Evolution Bug Severity and Priority Prediction.
Supervisors : Dr. V. B. Singh and Dr. Meera Sharma
Th24990

Abstract
(Not Verified)

Software repositories are generated during development of software projects and contain a lot of valuable facts. Using the information stored in these repositories, researchers need not to depend on their own intuition or experience, but may rely on historical data and field data. Researchers analyze, develop prediction models and cross-link with the wealth of data available in the software repositories to discover interesting and useful information about software systems and projects. A variety of open source software repositories are available, namely “source control repositories”, “bug repositories”, “archived communications”, “distribution logs” and “code repositories”.The bug report data is managed using bug reporting and tracking systems (“BugZilla”, “Jira”, “Mantis”, “Trac”, “Gnats”, “Fossil” and “Bugtracker.net”) separately. Bug report data is useful in various bug attributes prediction. This bug report data is analyzed by bug triager to resolve different bugs for the maintenance and evolution of the software products. Triager is the person who analyzes and refines the bugs by using his knowledge and experience. Bug triaging is an important part of the bug resolution process. A bug report is characterized by various attributes. Some are assigned at the initial time of bug reporting and some are assigned during the fixing process of the bugs. In order to predict the other attributes and to improve the quality of software, a clear understanding of bug attributes, their contribution, their interconnection, and their interdependence need to be understood. Some attributes have textual description such as “bug summary” and “long description”. Some attributes are discrete in nature for example, “bug_id”, “platform”, “number of CC”, “assignee”, “operating system”, “hardware”, “component”, “reporter”, “resolution”, “product”, “status”, “severity” and “priority”. Bugs are reported on the bug tracking system by different users with a fast speed. The size of software repositories is also increasing with enormous rate. This increased size often has much uncertainty and irregularities. The factors that cause uncertainty are biases, noise and abnormality in data. We consider that software bug report phenomena on the bug tracking system keep an irregular state.Without proper handling of these uncertainties and irregularities, the performance of learning strategies can be significantly reduced. These uncertainties and irregularities in bug report data can be measured in terms of entropy. Open source software evolves with the active participation of users in terms of reporting different issues, namely “bugs”, “new feature requests (NF)”, and “feature improvements (IMP)”. These active users are distributed in different geographical regions and are working for the evolution of open source software. Code changes are performed in source code files due to “bug fixes”, “new features”, and “feature improvements” and increase “code complexity” / “randomness” / “uncertainty” in source code files. Clearly, open source software will evolve through these code changes, and a clear understanding of “bug fixing”, “new features”, “improving existing features” and changes in source code files is required. Changes to source code introduced in the software to fix various issues are quantified using “entropy-based measures” and named as “complexity of code changes”.Within the current release some of the issues stay unresolved. These unresolved / leftover issues are added to the next release 's initial issue content, and are

fixed in subsequent releases.Bug severity indicates the impact of the bug on the functionality of the software or its components. In bug repositories, severity is labeled as “Blocker”, “Critical”, “Major”, “Normal”, “Minor”, “Trivial” and “Enhancement”. Due to lack of awareness and knowledge about the software, people make mistakes in assigning level of severity at the time of bug reporting. Manual identification and verification of bug severity is a tedious and time-consuming task. Bug priority determines the importance of a bug in the presence of other bugs. Bugs are prioritized by P1 level, i.e. the most important to P5 level, i.e. the least important. The correct assignment of bug priority and severity helps in “bug fix scheduling”/ “assignment” and “resource allocation”. Incorrect allocation of severity and priority levels delay the resolution of important bugs and can result in inefficient use of the resources (wasting time and effort by identifying unimportant bugs first). In order to correctly identify the severity and priority of a bug we require automation of bug priority and severity prediction. In literature, researchers have made attempts for bug summary based severity and priority prediction. But no attempt has been made to handle uncertainty in bug summary for bug severity and priority prediction.The validation of cross project is a key concern in empirical software engineering where we train the classifiers with historical data of projects other than the testing projects. The entropy-based measure can be been used to predict bug severity and priority of newly coming bug reports in cross project context.We assume that the bug reports, i.e. different bug attributes, reported in software bug repositories are trustworthy during bug triaging process. In reality, the bug reports data is not trustworthy in terms of various aspects like integrity, authenticity and trusted origin as the bugs are reported by users who may or may not have proper knowledge of the software. It may result in uncertainty in reported bug data. Without proper handling of these uncertainties in different bug attributes, the performance of learning strategies used for different bug attributes prediction can be significantly reduced. In order to consider these uncertainties, entropy based measure can be used to build prediction models. Veracity refers the quality of the data (e.g., “uncertain” or “imprecise data”). Because bug report data can be “uncertain”, “inconsistent”, “noisy”, “ambiguous”, or “incomplete”, data veracity is an important issue that needs to be addressed. Models are required to estimate veracity (trustworthiness) of bug reports data from the information hidden in the data itself. We have measured the uncertainty by using Shannon Entropy. “If the entropy score is high, then there will be a higher degree of ambiguity which shows less certainty and less veracity. If the entropy score is low, it will show a high degree of certainty and thus higher degree of veracity. It means that lower the information entropy higher the veracity. Higher the information entropy value, lower the veracity.”

Contents

06. RAKHI

Relevance Based Statistical Analysis of Multiword Expressions for Hindi Novels.

Supervisor : Dr. Archana Singal

Th 24989

Abstract
(Verified)

Multiword Expressions (MWEs) are one of the important aspects of text processing which is used to find the correct meaning of a text phrase. Due to the idiomatic nature of MWEs and their frequent occurrence in all types of text, there may be some difficulty in identifying and extracting various types. This can be solved by using the statistical methods and thus will help in many Natural Language Processing (NLP) applications. The proposed work mainly focuses on the issues of recognizing and analyzing MWEs for Hindi, as Hindi MWEs did not get much

attention from the earlier researchers. A benchmark Hindi dataset has been collected from Hindi novels “Godan”, “Karambhumi” and “Alankar” written by “Munshi Premchand Ji”. Some new types of MWEs which were ignored by earlier researchers have also been identified and a standard classification has been suggested in the form of formal constructions and functional classes of Hindi MWEs. An ontological representation has also been given for the proposed classification of Hindi MWEs. The classification process has been further improved by Support Vector Machine (SVM) using the feature vector. In the proposed work, statistical analysis has also been performed for Hindi MWES. The statistical aspects have been discussed in the form of baseline and statistical measures. The baseline measures are used to evaluate the performance of the system and to find the relation among various statistical measures. The statistical measures are based on the frequency of occurrence of a particular word pattern in a corpus. The existing, as well as proposed baseline and statistical measures, have been evaluated using the multiple threshold method for efficient evaluation of corpus size.

Contents