International Journal of Combined Research & Development (IJCRD)
eISSN:2321-225X;pISSN:2321-2241 Volume: 2; Issue: 2; February-2014

RANDOM DATA PERTURBATION TECHNIQUE ON MODEL BASED COLLABORATIVE TECHNIQUE USING COMPOSITE PROTOTYPE METHOD

THOMURTHY. Murali Mohan.[1], KOICHI Harada[2], Balakrishna.ANNEPU[3]
1. Corresponding Author:thomurthy@yahoo.com, India.
2. Professor, Department of Engineering, Hiroshima University, Hiroshima, Japan.
3. Assistant Prof. Noble Institute of Science and Technology. India.

Abstract:

Number of approaches which use Model-based collaborative Filtering (MBCF) for scalability in buildings recommendation systems in web personalization have poor accuracy due to the fact that web usage data is often sparse and noisy. In this papers the basic concept of model-based collaborative filtering systems and the most popular algorithms- Apriori algorithm, Simple CF Algorithm and singular value Decomposition algorithm techniques and there importance' are discusses a new model-based collaborative filtering algorithm based on composite prototype is proposed by introducing modifications in singular value decomposition technique. The methodology using composite prototypes used for predictions have been discussed. The Formulas that were used to implement these models including Rank determination, gradient, derivatives, Frobenius form and Prediction. The measured Mean Absolute Error (MAE) of the proposed model is compared with available models and finally the performance analysis is done based on parameter MAE.

Keywords: Recommender Systems, Bayesian network, Apriori algorithm, SVD, Mixture Models.

1. Introduction

Model-based collaborative filtering algorithms based on composite prototypes allow the system to learn to recognize complex patterns based on the training data, and then make intelligent predictions for the collaborative filtering tasks for test data or real-world data, based on the learned models. Clustering [12], mining association rules, and sequence pattern discovery have been used to determine the access behavior model. Making use of some of the characteristics of the modeling process can provide significant improvements to recommendation effectiveness. Model-based CF algorithms[11] have been investigated to solve the shortcomings of memory-based CF algorithms. Usually, classification algorithms can be used as CF models if the user ratings [1] are categorical, and regression models and SVD methods and be used for numerical ratings. In section 1 explains the introduction of the part of the work, section 2 is related work, section 3 & 4 proposed model and algorithm, section 5 implementation, section 6 model Experimentation, section 7 results and discussion, section 8 end of this work with conclusion.

2. Related Work

Content-based filtering [9] and retrieval builds on the fundamental assumption that users are able to formulate queries that express their interests or information needs in term of intrinsic features of the items sought. Model based Collaborative filtering is a technology that is complementary to content based filtering and that aims at learning predictive models of user preferences, interests or behavior from community data, that is, a database of available user preferences. Virtually all first generation recommender systems[8] [2] have used the same

fundamental two-step approach of first identifying users that are similar to some active user for which a recommendation has to be made, and then computing predictions and recommendations based[5] on the preferences and judgments of these similar or like-minded users. Model based collaborative techniques [4][6] are classified into three categories and they are

1. Bayesian Belief Net CF Algorithm
2. Apriori Algorithm
3. Singular Value Decomposition Algorithm.

2.1 Bayesian Belief Net CF Algorithm:

This network is a graphical representation of the joint probability distribution for a set of variables. This representation was originally designed to encode the uncertain knowledge [3] of an expert. They also have become the representation of choice among researchers interested in uncertainty in Artificial Intelligence. The first is a Bayesian belief net (BN) [10] is a directed, acyclic graph (DAG) with a triplet N,A,O, where each node n ∈ N represents a random variable, each directed arc a ∈ A between nodes is a probabilistic association between variables, and O is a conditional probability table quantifying how much a node depends on its parents. The second component is a collection of local interaction models that describe the conditional probability of each variable X, given its parents. These two components represent a unique joint probability distribution over the complete set of variables. To distinguish items, transformed numerical ratings into these two labels are considered. To avoid this problem, we use F-Measure, which combines Precision and Recall:

$$\text{F-Measure} = \frac{2.\text{Precision.recall}}{\text{Precision+recall}}$$

a. Transformed data model results

Number of	Classification of

Features	accuracy (%)
50	64.0
100	65.0
150	66.0
200	67.0
250	66.9
300	66.8
350	66.7
400	66.6
450	66.5
500	66.4
550	66.3

Table 1. Classification of accuracy – Transformed data model.

b. Sparse data model

Number of users	Classification of accuracy (%)
10	66.0
20	66.5
30	67.0
40	67.5
50	68.0
60	67.75
70	67.50
80	67.25
90	67.00
100	66.50
110	66.00
120	65.50
130	65.00
140	64.50
150	64.00
160	63.50

Table.2 Classification of accuracy – Sparse data model

2.2 Apriori algorithm:

In this algorithm the rules are "if-then rules" with two measures which quantify the support and confidence of the rule for a given data set. The first and possibly most influential algorithm for efficient association rule discovery is Apriori. Association rule mining and its association rules can find out the predefined minimum support and confidence from a given database. The problem

is usually decomposed into two sub problems. One is to find those itemsets[7] whose occurrence exceeds a predefined threshold in the database; those itemsets are called frequent or large itemsets. The second problem is to generate association rules from those large itemsets with constrain of minimal confidence. the association rule recommender does make a more general prediction; it predicts a binary "like" or "dislike" classification for a recommended item if the confidence value is positive or negative, respectively. The resulting association rules are

Rules	Support (XY)	Support (Y)	Confidence
{A}=>{C}	2.0	2.0	1.0
{C}=>{A}	2.0	3.0	0.6666666666666666
{B}=>{C}	2.0	3.0	0.6666666666666666
{C}=>{B}	2.0	3.0	0.6666666666666666
{B}=>{E}	3.0	3.0	1.0
{E}=>{B}	3.0	3.0	1.0
{C}=>{E}	2.0	3.0	0.6666666666666666
{E}=>{C}	2.0	3.0	0.6666666666666666
{B}=>{C E}	2.0	3.0	0.6666666666666666
{C E}=>{B}	2.0	2.0	1.0
{C}=>{B E}	2.0	3.0	0.6666666666666666
{B E}=>{C}	2.0	3.0	0.6666666666666666
{E}=>{B C}	2.0	3.0	0.6666666666666666

Table 3. Strong association rules from the frequent itemsets.

2.3 Single Value Decomposition

The singular value decomposition (SVD) plays a vital role in numerical linear algebra and in many statistical techniques as well. Using two ortho normal matrices, SVD can diagonalizable any matrix A and the results of SVD can notify a lot about consequences of the matrix. A collaborative filtering deals with a large sized matrix which stands for customers and items. . Singular Value Decomposition states that every matrix A_{mxn} can be decomposed as

$$A = USV^T,$$

Where U and V are orthogonal and S is diagonal with singular values of A on the diagonal.

U, S and V values are maximum in full singular value decomposition. The MAE values are computed using existing Singular value decomposition (SVD) algorithm and modified SVD for test data sets are tabulated.

Neighbor Set Size	4	8	12	16	20	24	28
MAE For SVD	1.086	1.086	1.086	1.086	1.086	1.086	1.086

Table 4. MAE values for different neighbor Sets datasets

From the above analysis and observations Singular Value Decomposition is then best method for prediction. That is the reason we try to make some modification in SVD approach using Composite Prototypes as the proposed model and results are compared with the existing SVD model in the next sections.

3. Proposed Model

(Methodology of Model-Based Collaborative Filtering Algorithm Based On Composite Prototypes)

Given a matrix R, Compute a rank-r R_{app} (approximation) to this matrix such that the Frobenius form of $R - R_{app}$ is minimized. Then, Frobenius form ($||R - R_{app}||F$) is defined as simply the sum of squares of elements in $R - R_{app}$. It can achieve such an approximation by only considering the first r most significant singular values in the singular value decomposition of R. Returning to our domain, it can formulate the problem as an **u x m** matrix R which contains the actual ratings by the users, where u is the number of users and m is the number of movies. Assume that consider f features, regarding the rest as insignificant. Compute an approximation R_{app} to this matrix R, such that $||R-R_{app}||F$ is minimized, and $R_{app} = P_{uxf}(F_{mxf})T$. Notice that the i^{th} row of P vector is the preference vector for user i, and the k^{th} row of F is the feature vector for movie k. Therefore we have extracted the an approximation to the desired data, which can then use to fill unknown entries of R by computing the dot product of user preference and movie feature vectors. First of all,

it is by itself a difficult task to compute & rate features for each movie due to the subjective nature of the task. Second, this would require retrieving information from external resources and combining it with the user-movie rating and this data would require tremendous cleaning-up effort.

4. Proposed Algorithm

Algorithm: Require: average ratings. the given ratings convert into a matrix of ratings R, Compute an approximate matrix R_{app} such that MAE is minimized.

Step1: Task: Find the best dictionary to represent the data samples as sparse compositions
Step 2: Initialization: Set the dictionary matrix D. Set J = 1. Repeat until convergence
Step 3: Sparse Coding Stage - Use any pursuit algorithm to compute the representation vectors.
Step 4: Update Stage- For each column k = 1; 2; : : : ;K in D
Step 5: Compute the overall representation error matrix
Step 6: Restrict E by choosing only the columns corresponding to k**.**
Step 7: Apply SVD decomposition. Choose the updated dictionary column.
Step 8: Update the coefficient vector multiplied vectors

$$MAE = ||R-R_{app}||F$$

Step 9: compute P as $US1^{/2}$ and F as $S^{1/2}V'^{T}$
Step 10: minimize the error: $E = (R-R_{app})_{ij}^{2}$
Step 11: compute Pik (t+1) and F_{jk}(t+1) Take the derivative with respect to p_{ij} and f_{jk} and the updates become:

$$P_{ik}(t+1) = P_{ik}(t) + L*(R-Rapp)_{ij}* F_{jk}(t) – K* P_{ik}(t)$$

$$F_{jk}(t+1) = F_{jk}(t)+ L*(R-Rapp)_{ij} * P_{ik}– K* P_{ik}(t)$$

5. Implementation

The implementation of the proposed model is done using JAVA. The description of implementation process is as follows: The main java classes designed and developed to evaluated the predictions for the SVD Filtering algorithms are *CBA5.java, NBSSimblanceRow.java, Probability.java YSplineRendererDemoTest.java* .A segment of java code snippet and the structure of the java classes that implements the SVD Filtering algorithms proposed in the system are as follows.

```
List original = new ArrayList ();
String fileName2 = "D:\\Excelwork\\ml-
data_0\\u.data";
int usersSize = 100;
int itemsSize = 1000;
we.initialize(original, usersSize, itemsSize);
we.populateFileToList(original, fileName2,
usersSize, itemsSize);
```

Here the List '*original*' is the list which contains the original ratings of the users which will be compared with the predicted ratings. It is designed to populate the list with the ratings read from the u.data file with the mentioned Path in the code.

```
List test = new ArrayList();
fileName2 = "D:\\Excelwork\\ml-
data_0\\u5.test";
we.initialize(test, usersSize, itemsSize);
we.populateFileToList(test, fileName2,
usersSize, itemsSize);
```

The List '*test*' is the list which contains the test ratings of the users. Test data is the subset of original data. Using test data, it is designed to produce the user rating predictions and populating the 'test' list with values read from u5.test.

```
fileName2 = "D:\\Excelwork\\ml-
data_0\\u.genre";
ArrayList genre = new ArrayList();
we.initializeGenre(genre, fileName2);
```

The List '*genre*' is the list of genre of the movies. Each movie genre is given a unique number which is used in item classification and populating the test list with values read from u5.test.

```
fileName2 = "D:\\Excelwork\\ml-
data_0\\u.item";
List items = new ArrayList();
we.initializeItems(items, 1682, 30);
```

we.populateItemsToList(items, fileName2, 1682, 30, genre);

The List 'items' is the list of all the items that presented in u.item. 1682 is number of items given, and 30 is the number of properties mentioned in the u.item file. It is designed and developed to populate the test list with values read from u.item. All the properties are embedded in a child list and the child list is added to parent list. Here a list s3 is generated, which contains user given ratings along with content boosted predicted values filled in the place of non given ratings.

```
for(int user=0; user<s3.size(); user++){
currentUserData = (ArrayList)s3.get(user);
for(int item=0; item<currentUserData.size();
item++){ intRating = (Double)
currentUserData.get(item);
 rating = intRating.doubleValue();
 if(rating == 0 )
{rating = avgRating;}
ratingsVector [user][item] = rating; } }
```

Here *ratingsVector* has been generated, which is a double indexed array contains user ratings along with default values generated by SVD collaborative filtering algorithm.

6. Model Experimentation

Dataset description: One of the largest datasets of explicit user preferences is MovieLens, a movie rating database collected over a period of 18 months by the Compaq Corporation. EachMovie contains the ratings of approximately 60,000 users for a set of 1,800 movies, 2.8 million ratings in all, or an average of 46 ratings per user. The rating scale ranges from 0 to 5. A subset of the ratings data from the MovieLens data set used for the purposes of comparison. 20% of the users were randomly selected to be the test users. In the dataset from grouplens website [114], it mentioned that the data sets u1.base and u1.test through u5.base and u5.test are 80%/20% splits of the u data into training and test data. Each of u1, u2, u3, u4, and u5 has disjointed test sets for cross validation. These data sets can be generated from u.data by mku.sh. Source file u.data contained the u dataset by 943 users with 100000 ratings on 1682 items. Each user has rating at least 20 movies. This is a tab separated list of user id, item id, rating and timestamp.

7. Results & discussion

The MAE values are computed using existing Singular value decomposition (SVD) algorithm and modified SVD for different test data sets u1.test, u2.test, u3.test, u4.test and u5.test and tabulated in table 5 to table 9. The Comparative analysis of these computed values are presented.

a) MAE values for SVD on U1.test dataset

Neighbor Set Size	4	8	12	16	20	24	28
MAE for SVD Existing	1.086	1.086	1.086	1.086	1.086	1.086	1.086
MAE for SVD Modified	1.049	1.049	1.049	1.049	1.049	1.049	1.049

Table 5: MAE Values for different neighbor sets for CF on U1.Set

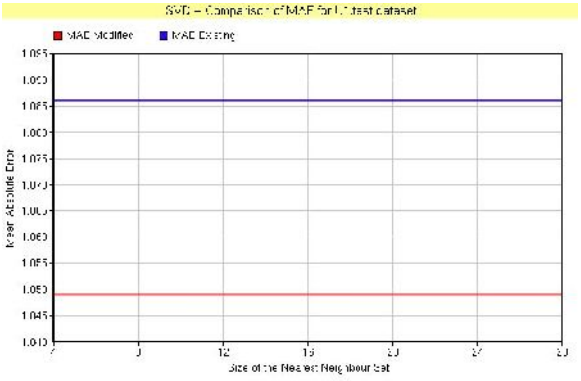

Figure 1.Comparison of MAE for Singular Value Decomposition (SVD) algorithm vs proposed algorithm on the U1.test dataset

MAE is shown in as two graphical representations, the blue line, represents an existing Singular Value Decomposition and the red line, and represents a modified algorithm, with lesser values than the existing.

b) MAE values for SVD on U2.test dataset

Neighbor Set Size	4	8	12	16	20	24	28
MAE for SVD Existing	1.110	1.110	1.110	1.110	1.110	1.109	1.109

MAE for SVD Modified	1.091	1.090	1.090	1.090	1.090	1.090	1.090

Table 6: MAE values for different neighbor sets for CF on u2.test

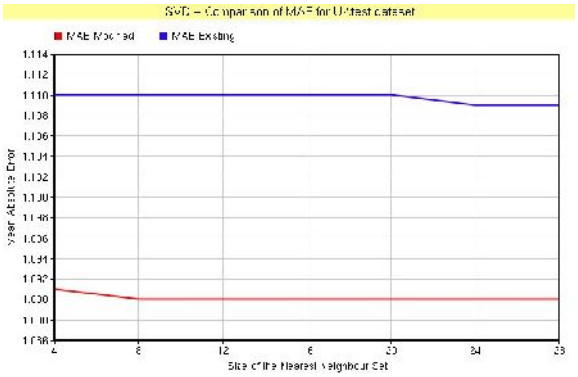

Fig. 2. Comparison of MAE for Singular Value Decomposition (SVD) algorithm vs proposed algorithm on the U2.test.

C) MAE values for SVD on U3.test dataset

Neighbor Set Size	4	8	12	16	20	24	28
MAE for SVD Existing	1.110	1.110	1.110	1.110	1.110	1.110	1.110
MAE for SVD Modified	1.091	1.091	1.091	1.091	1.091	1.091	1.091

Table 7: MAE values for different neighbor sets for CF on u3.test

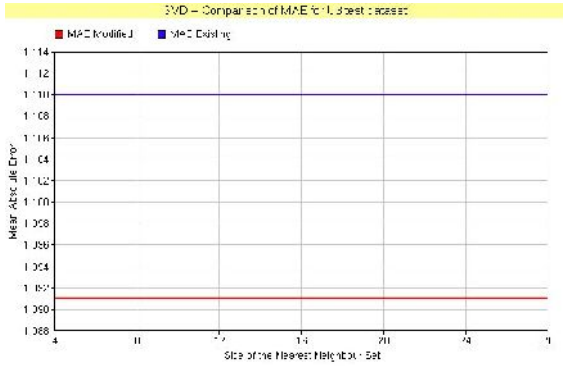

Fig.3. Comparison of MAE for Singular Value Decomposition (SVD) algorithm vs proposed algorithm on the U3.test.

D) MAE values for SVD on U4.test dataset

Neighbor Set Size	4	8	12	16	20	24	28
MAE for SVD Existing	1.210	1.210	1.210	1.210	1.210	1.210	1.210
MAE for SVD Modified	1.161	1.161	1.161	1.161	1.161	1.161	1.161

Table 8 : MAE values for different neighbor sets for CF on u4.test

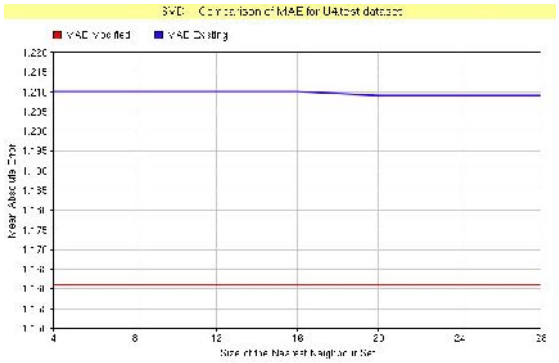

Fig.4. Comparison of MAE for Singular Value Decomposition (SVD) algorithm vs proposed algorithm on the U4.test.

E) MAE values for SVD on U5.test dataset

Neighbor Set Size	4	8	12	16	20	24	28
MAE for SVD Existing	1.187	1.187	1.187	1.187	1.187	1.187	1.187
MAE for SVD Modified	1.168	1.168	1.168	1.168	1.168	1.168	1.168

Table 9: MAE values for different neighbor sets for CF on u5.test

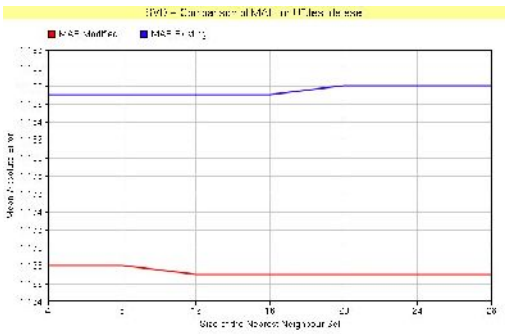

Fig.5. Comparison of MAE for Singular Value Decomposition (SVD) algorithm vs modified algorithm on the U5.test

The results presented in this chapter are given according to evaluation procedures with the

experiments performed. The results for existing incremental SVD and the modified will be compared and presented. Derived MAE values for different test datasets from U1.test to U5.test is related with recommendation accuracy which is computed and compared for the existing and modified methods to see which one performs better. As mentioned earlier of this section, the dataset and the evaluation metric, mean absolute error (MAE) is evaluated for every fold in our 5-fold cross validation experiment. Finally the total MAE was computed from the whole set of users and folds in the experiments. The MAEs for the different NNSs evaluation using U1.test dataset performs only 3.41% better improvement over existing SVD. Whereas with U2.test dataset and U3.test dataset it is only 1.71% increase is observed. 4.04% improvement is noticed in case of the results performed with U4.test dataset. 1.90% of improvement is noticed in U5.test dataset. It can be observe that both methods are the performed with unique performance in most of the cases the different in improvement is also very low. The prediction quality is decreased with increase of NNS in many cases. The overall performance of the modified SVD is slightly better than the existing.

8. Conclusion

Simple Bayesian classifier, Apriori and Singular value decomposition algorithms are implemented in this chapter but only the singular value decomposition has been selected for modification as it is found to generate quick information needed apart from delivering high level pattern comparison. Overall the implemented algorithms i.e., Simple Bayesian CF Algorithm, Apriori algorithm and singular value decomposition (SVD), singular value decomposition algorithm performed well while deriving the prediction quality with Mean Absolute Error (MAE). The modified version results of incremental SVD are compared with the existing SVD algorithm. The modified incremental SVD method algorithm performs slightly well when compared to the standard algorithms when singular values are isolated in terms of prediction quality by performance evaluation of MAE. Experimenting with entirely different algorithms and combining results seems to be the best approach to improve particularly in model-based collaborative filtering.

9. Bibliography

[1]. D. Agarwal, A. Broder, D. Chakrabarti, D. Diklic, V. Josifovski, and M. Sayyadian. Estimating rates of rare events at multiple resolutions. *Procs of the 13th ACM SIGKDD*, pages 16–25, 2007.

[2].R. Bell, Y. Koren, and C. Volinsky. Modeling relationships at multiple scales to improve accuracy of large recommender systems. *Procs of the 13th ACM SIGKDD*, 2007.

[3]. A. Gelman. *Bayesian Data Analysis*. CRC Press, 2004.

[4].G. Linden, B. Smith, and J. York. Amazon. com Recommendations: Item-to-Item Collaborative Filtering. 2003.

[5]. B. Sarwar, G. Karypis, J. Konstan, and J. Reidl. Item-based collaborative filtering recommendation algorithms. *Procs of the 10th int. conference on World Wide Web*, 2001.

[6]. G. Linden, B. Smith and J. York, “Amazon.com Recommendations: Item-to-item Collaborative Filtering", *IEEE Internet Computing* **7** (2003), 76–80.

[7]. J.Wang, A.P. de Vries and M. J. T. Reinders,“Unifying User-based and Item-based Collaborative Filtering Approaches by Similarity Fusion", *Proc. 29th ACM SIGIR Conference on Information Retrieval*, pp. 501–508, 2006.

[8]. G. Adomavicius and A. Tuzhilin, “Towards the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions", *IEEE* Transactions *on Knowledge and Data Engineering* **17** (2005), 634–749.

[9]. BASU, C., HIRSH, H., AND COHEN, W. 1998. Recommendation as classification: Using social and content-based information in recommendation. In *Proceedings of the Recommender SystemWorkshop*.11–15.

[10]. CHIEN, Y.-H. AND GEORGE, E. 1999. A Bayesian model for collaborative filtering. In *Online Proceedings of the Seventh International Workshop on Artificial Intelligence and Statistics*.

[11]. HECKERMAN, D., CHICKERING, D. M.,MEEK, C., ROUNTHWAITE, R., AND KADIE, C.M. 2000. Dependency networks for inference, collaborative filtering, and data visualization. *J. Mach. Learn. Res. 1*, 49–75.

[12]. UNGAR, L. AND FOSTER, D. 1998. Clustering methods for collaborative filtering. In *Proceedings of the Workshop on Recommendation Systems*. AAAI Press, Menlo Park, Calif.

AUTHOR'S PROFILE

Thomurthy Murali Mohan is a Assistant Professor from Kaushik College of Engineering, affiliated to Jawaharlal Nehru Technological University. He received the B.Sc (Computer Science), M.Sc (Computer Science) in 2003 & 2005 from Andhra University. He Completed M.Tech (Computer Science & Engineering) in 2012 from Jawaharlal Nehru Technology University, Kakinada. He received MBA in 2010 from Punjab Technical University. His current research is mainly in Data Mining, and Data Warehousing. Special interests include in Robotics.

Koichi Harada is a professor of the graduate Schools of Engineering at Hiroshima University. He received the BE in 1973 from Hiroshima University, and MS and Phd in 1975 and 1978, respectively, from Tokyo Institute of Technology. His current research is mainly in the area of the computer graphics. Special interests include man-machine interface through graphics; 3D data input techniques, data conversion between 2D and 3D geometry, effective interactive usage of curved surfaces. He is a member of ACM, IPS of Japan, and IEICE of Japan

Annepu Balakrishna is a professor from Noble Institute of Engineering college, Visakhapatnam. He received MCA in 2009 from Andhra University, and M.Tech in 2013 from JNT University, Kakinada and He Ph.D in 2012 from CMJ University. His area of interstesa are Software Engineering. His current research is mainly in Data Mining and Data Ware Housing. Special interestes include in Robotics.