International Journal of Engineering Applied Sciences and Technology, 2020
Vol. 5, Issue 2, ISSN No. 2455-2143, Pages 288-293
Published Online June 2020 in IJEAST (http://www.ijeast.com)

RECOVERY OF MISSING DATA USING ENHANCED PROBABILISTIC MATRIX FACTORIZATION

U. Priyanka
M.E., Computer Science and Engineering
Government College of Technology
Coimbatore -641 013

Dr. K. Kumar
M.E., PhD, Computer Science and Engineering
Government College of Technology
Coimbatore -641 013

ABSTRACT: Recommendation systems are one of the most widespread forms of machine learning in modern society. Whether you are looking for your next show to watch on Netflix or listening to an automated music playlist on Spotify, recommender systems impact almost all aspects of the modern user experience. One of the most common ways to build a recommendation system is with matrix factorization, which finds ways to predict a user's rating for a specific product based on previous ratings and other users' preferences. In this project, we compare and contrast several PMF-based models by applying them to find missing values in music recommendation system. Motivated by the observation that incorporating user network information is not as effective as constraining the user feature vector with latent constraint similarity matrix, we developed Constrained Kernelized PMF (cKPMF) model. We show that cKPMF is the most effective model for our task at hand among the models explored in this project. In this article we formulate the missing value estimation as a recommender system problem.

Keywords: Missing Data Prediction, Recommendation System, Music Dataset, Probabilistic Matrix Factorization, Kernalized PMF, Matrix Factorization

I. INTRODUCTION

Missing data is the most common problem when analyzing data in real world., missing data is present to various applications ranging from gene analysis to sensor applications. As many applications and machine learning algorithms rely on complete data sets, it is most important to handle the missing data appropriately. Missing value recovery is an important part of data preprocessing.

In some cases, simple approaches may handle missing data. For example, complete-case analysis uses only the known data and omits all unknown and null observations to conduct statistical analysis. It works well if only a few observations are missing, and when the data is missing completely at random, complete case analysis does not lead to biased results (Little and Rubin, 1987). Alternately, some machine learning algorithms is used for missing data, and there is no need for preprocessing. For instance, CART and K-means have been used for data with missing values. (Breiman et al., 1984; Wagstaff, 2004).

In many other situations, missing values need to be recovered prior to running statistical analyses on the complete data set. The benefit of this approach is that once a complete data has been generated, many learning algorithms can be applied to the imputed data set. The objective is to impute values resemble to the complete data as close as possible.

Collaborative filtering(CF) is an effective way to implement recommender system (Ekstrand et al., 2011). Recommending values for missing data using collaborative filtering. Dealing with sparse and imbalanced data and to be able to scale large dataset are the two key challenges in developing CF models.

Probabilistic Matrix Factorization(PMF) introduced by (Mnih & Salakhutdinov, 2008) decomposes the matrix into product of two matrices through factorization and it has been flexible and effective framework to address large, sparse and very imbalanced dataset. Constrained Probabilistic Matrix Factorization (cPMF) is proposed as a variation of the simple PMF model. Kernelized Probabilistic Matrix Factorization(KPMF) model introduced by (Zhou et al., 2012) is able to effectively incorporate information from user and item to improve recommender's performance.

II. EXISTING TECHNIQUES

MISSING DATA IMPUTATION

Fang et al., (2014) extend PMF by decomposing trust information into four general trust aspects, i.e. benevolence, integrity, competence, and predictability, and incorporate them into the PMF model with support vector regression. (Salakhutdinov & Mnih, 2008) propose a Bayesian PMF, which generalizes PMF to handle non zero mean and non-spherical Gaussian priors. The advantage of BPMF is that it is less prone to overfitting, however it suffers from high computing complexity. In the context of music recommendation where model scalability is an important issue, it is often not the most ideal choice.

International Journal of Engineering Applied Sciences and Technology, 2020
Vol. 5, Issue 2, ISSN No. 2455-2143, Pages 288-293
Published Online June 2020 in IJEAST (http://www.ijeast.com)

Halatchev et al., (2005) proposed the basket association rule mining technique to the data stream environment through Data Stream Association Rule Mining (DSARM). It utilizes Window Association Rule Mining(WARM). These algorithms are subject to two response time constraints-soft and hard deadline constraints framework.

Gruenwald et al., (2007) proposed a data estimation technique using association rule mining on stream data based on closed frequent itemsets (CARM) to discover relationships between sensors and use them to compensate for missing data. Estimation accuracy and both time and space efficiency can be improved.

Gruenwald et al., (2007) a data mining based technique, called Freshness Association Rule Mining (FARM) to estimate values for missing, corrupted, or late readings from one or more sensors in a sensornet at any given round.

Ruslan Salakhutdinov et al., (2008) proposed Bayesian treatment of the Probabilistic Matrix Factorization (PMF) which can be trained using Markov chain Monte Carlo (MCMC) methods for approximate inference in this model. it allows the confidence in the prediction to be quantified and taken into account when making recommendations using the model.

YuanYuan et al., (2008) use a hierarchical unsupervised fuzzy ART neural network to represent the data cluster prototypes. Spatial-temporal imputation technique is used to estimate the missing values. It performs better than other estimation algorithms including moving average and Expectation-Maximization (EM) imputation.

Linghe Kong et al., (2013) design a novel *environmental space time improved compressive sensing* (ESTI-CS) algorithm for estimating the missing data. ESTICS embeds customized features into baseline CS to deal with the specific data loss patterns, which computes the minimal low-rank approximations of the incomplete EM and refines the interpolation with spatio-temporal features.

Fang et al., (2014) extend PMF by decomposing trust information into four general trust aspects, i.e. benevolence, integrity, competence, and predictability, and incorporate them into the PMF model with support vector regression. It considers the association between trust and the latent user feature matrix. This is accordance with social influence theory that a user will become more similar to other users trusting and being trusted by the user.

Fekade et al., (2017) proposed recovery of missing data through probabilistic matrix factoristion. k-means algorithm is used to cluster the sensor data and through factorization, missing data has been recovered.

Dimitris et al., propose a family of new imputation methods, opt. impute, which finds high quality solutions to this problem using fast firstorder methods. Through extensive computational experiments on 84 data sets from the UCI Machine Learning Repository, we show that opt. impute yields statistically significant gains in imputation quality over state-of-the-art imputation methods, which leads to improved out-of-sample performance on downstream tasks

III. PROPOSED METHODOLOGY

A. KERNALISED PROBABILISTIC MATRIX FACTORISATION

Recall that in the PMF models, rows of U and V are assumed to be independent, and we are only using the rating matrix as input. The Kernelized probabilistic Matrix Factorization model allows U and V to capture the covariances between any 2 rows of U and V by assuming the columns of U and V are generated from a zero-mean Gaussian Process(GP). By generating the covariance matrices from user and item side information, we can easily incorporate them into the model. It showed that incorporating user network information with KPMF is very effective at improving prediction accuracy. However, in our context, the improvements were minimal. Meanwhile, constraining the user latent matrix with cPMF as well as incorporating item side information with KPMF was very effective. we constrain the user latent matrix as well as assuming Gaussian process distribution for the columns of item latent matrix.

The generative process for cKPMF is as follows (see figure

1. Generate $W_{k,:} \sim \mathrm{N}(0, \sigma^2 \boldsymbol{I})$ for $k \in \{1,...,M\}$
2. Generate $Y_{i,:} \sim \mathrm{N}(0, \sigma^2 \boldsymbol{I})$ for $i \in \{1,...,N\}$
3. Generate $V_{:,d} \sim GP(0, K_v)$ for $d \in \{1,...,D\}$
4. Generate indicator matrix I such that $I_{i,j} = 1$ if $R_{i,j}$ is observed, $I_{i,j} = 0$ otherwise.
5. For each non-missing entry $R_{i,j}$, generate $R_{i,j}$

$$\mathcal{N}((Y_{i,:} + \frac{\sum_{k=1}^{M} I_{i,k} W_{k,:}}{\sum_{k=1}^{M} I_{i,k}}) V_{j,:}^{T}, \sigma^2)$$

Notice that when Kv is a diagonal, cKPMF reduce to cPMF. The Stochastic Gradient Descent rules for Y and W are the same with the cPMF model, the update rule for V is as follows:

$$V_{j,:} := V_{j,:} + \alpha(err_{i,j} U_{i,:} - \frac{\sigma^2}{2\sum_{p=1}^{N} I_{p,j}} (\sum_{k=1}^{M} S_{vj,k} V_{k,:} + S_{vj,j} V_{j,:}))$$

Our experiment results show CT kernel outperforms the rest.

IV. DATASET

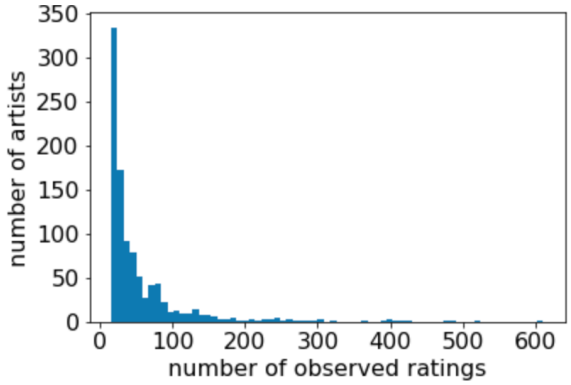
We are using hetrec2011-lastfm-2k, a set of social networking, tagging, and music artist listening information from Last.fm1 online music system. To speed up our experiments, we used a subset with the top 1000 most frequently rated artist. The statistics of the dataset is given in Table [1]. Observe that the rating density is very low. Figures shows the histogram for the number of ratings of each artist and each user. We observe that the

International Journal of Engineering Applied Sciences and Technology, 2020
Vol. 5, Issue 2, ISSN No. 2455-2143, Pages 288-293
Published Online June 2020 in IJEAST (http://www.ijeast.com)

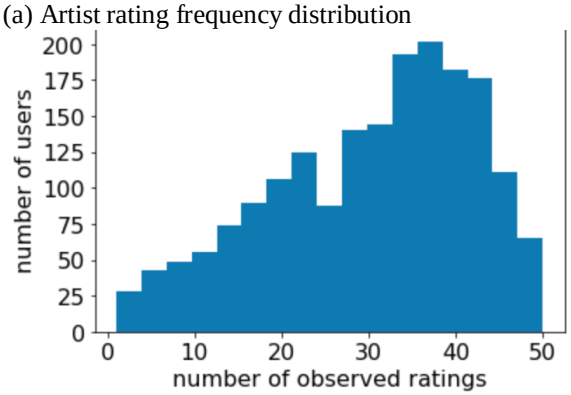
artist rating frequencies are more balanced, since most artists have similar rating frequencies. But user rating frequencies varies more significantly. So this dataset suffers from the typical sparsity and data imbalance problem.

ITEM	STATS
# USERS	1871
# ITEMS	1000
# RATINGS	56620
RATING DENSITY	3.03%
# RELATIONS	25424
# TAGS	87366

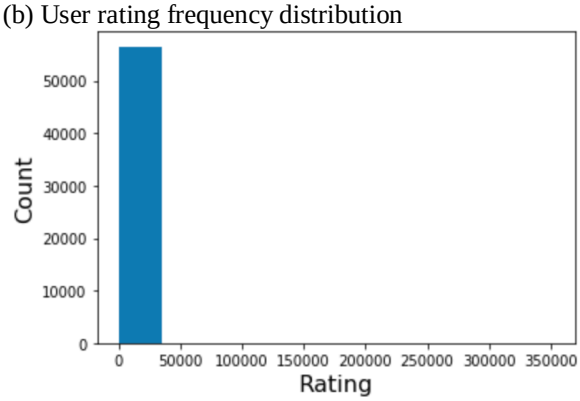
Table 1. Statistics of the dataset used.



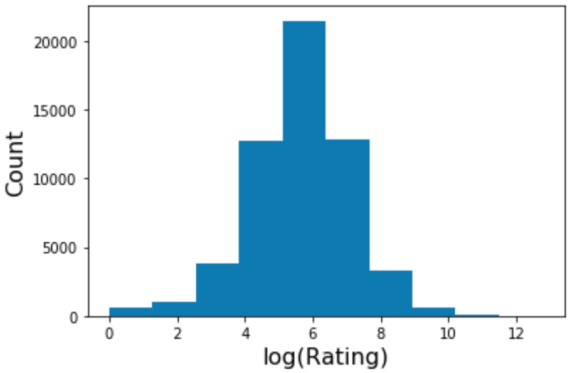
(a) Artist rating frequency distribution



(b) User rating frequency distribution



(a) Rating distribution



(b) Log-rating distribution

V. IMPLEMENTATION

Anaconda3

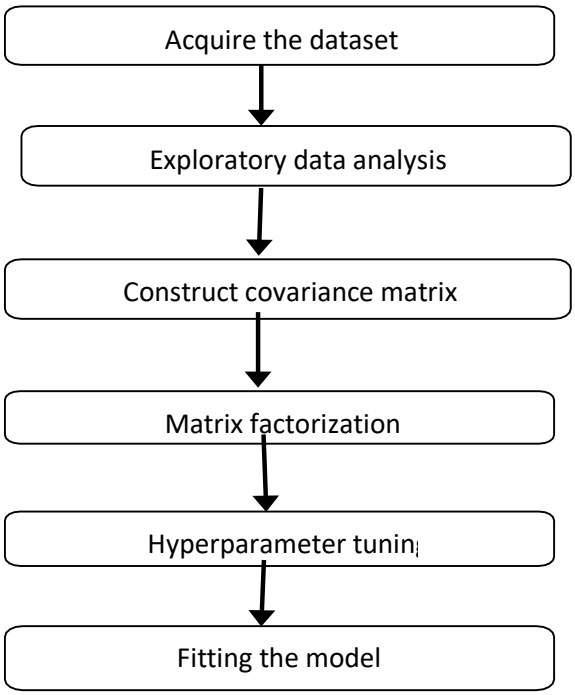
Anaconda is a free and open-source appropriation of the Python and R programming for logical figuring like information science, AI applications, large-scale information preparing, prescient investigation, and so forth. Anaconda accompanies in excess of 1,400 packages just as the Conda package and virtual environment director, called Anaconda Navigator, so it takes out the need to figure out how to introduce every library freely. Anaconda Navigator is a Graphical User Interface (GUI) incorporated into Anaconda appropriation, it enable the clients to dispatch applications and overview the conda packages, conditions, channels without utilize the command- line directions.

Python 3.7

Python is broadly utilized universally and is a high-level programming language. It was primarily introduced for prominence on code, and its language structure enables software engineers to express ideas in fewer lines of code. Python is a programming language that gives you a chance to work rapidly and coordinate frameworks more effectively.

International Journal of Engineering Applied Sciences and Technology, 2020
Vol. 5, Issue 2, ISSN No. 2455-2143, Pages 288-293
Published Online June 2020 in IJEAST (http://www.ijeast.com)

A. STRUCTURE OF THE MODEL



B. ACQUIRE THE DATASET

The dataset used in this paper is acquired from hetrec2011-lastfm-2k, a set of social networking, tagging, and music artist listening information from Last.fm1 online music system. To speed up our experiments, we used a subset with the top 1000 most frequently rated artist. We observe that the artist rating frequencies are more balanced, since most artists have similar rating frequencies. But user rating frequencies varies more significantly. So this dataset suffers from the typical sparsity and data imbalance problem.

C. EXPLORATORY DATA ANALYSIS

The sparsity of the dataset, or the number of known ratings in comparison to the number of possible user-item combinations is examined. Originally, only 0.2% of possible ratings within the data set were known, which is quite sparse and could negatively impact the accuracy of the model. Therefore, filter the data set to only include only users that had rated at least five artists. It increased the density of the data set to 1.3%, which, while still relatively sparse, will allow the matrix factorization model to make more accurate predictions. After these changes were made, there was one last change was to make before fitting the model. Within this data set, each user and artist is assigned a unique ID. However, in order to make using the recommendation model easier, make all of these IDs contiguous, such that they can be used to index into the embedding matrices.

Create a mapping, for both users and items, from the original ID to the new contiguous ID, such that all IDs fall within the range [0, total number of users/artists]. For easy look up and quick predictions perform this conversion for both users and artists ID into embedding matrices

D. CONSTRUCT COVARIANCE MATRICES

i) Construct Covariance Matrices for GP

A valid kernel function for GP should generate a covariance matrix that is positive semi-definite. There are many available choices (Hofmann et al., 2008).

a) CONSTRUCT Kv

To incorporate the artist side information into the covariance matrix, we first constructed a artist tag matrix using onehot encoding (figure 1(c)). We then applied the Radial Basis Function(RBF) kernel to the feature vectors to obtain Kv. The similarity measure between any 2 artists' feature vectors is calculated through RBF kernel.

b) CONSTRUCT Ku

Consider the users' social network as an undirected, unweighted graph G with nodes and edges representing users and their connections. We compared the 3 graph kernels described in (Zhou et al., 2012): Diffusion, Commute Time(CT), Regularized Laplacian.

In addition, we generated a node2vec (Grover & Leskovec, 2016) kernel matrix by first generating node embeddings with node2vec then convert the embedding matrix into a kernel matrix with RBF kernel.

The Kernelized Probabilistic Matrix Factorization model allows U and V to capture the covariances between any 2 rows of U and V by assuming the columns of U and V are generated from a zero-mean Gaussian Process(GP). By generating the covariance matrices from user and item side information, we can easily incorporate them into the model.

When the covariance matrices Ku and Kv are both diagonal, KPMF reduces to PMF. We are still assuming the independence between U and V and independence between the observed ratings.

E. MATRIX FACTORIZATION

We constrain the user latent matrix as well as assuming Gaussian process distribution for the columns of item latent matrix. The matrix factorization decomposed the original matrix into the product of two matrices by factorization. The factorized matrix gives the values of missing values. This often leads to overfitting problem. This problem can be solved through fixing the values of variance as constants.

F. HYPER-PARAMETER TUNING

Now that the data has been filtered and preprocessed, the recommendation model can actually be trained. However, training the model has a couple of hyper-parameters that must be set properly: the learning rate and the latent dimensionality. To determine the optimal learning rate, an adaptive learning rate selection technique is utilized. This technique trains the model for several iterations, increasing the learning rate used for

International Journal of Engineering Applied Sciences and Technology, 2020
Vol. 5, Issue 2, ISSN No. 2455-2143, Pages 288-293
Published Online June 2020 in IJEAST (http://www.ijeast.com)

updating the model's parameters on every iteration. The loss is then recorded for each iteration, the optimal initial learning rate is represented by the largest value for the learning rate before the loss begins to increase, which, in this case, was around 0.1. Therefore, the learning rate was initially set to 0.1 when the model was trained. Determining the optimal latent dimensionality was done through grid search.

G. FITTING THE MODEL

Now that the hyper-parameters have been selected, the model can be trained. Training was done three epochs at a time, and the learning rate was reduced by a factor of ~2 every three epochs until the model converged. By gradually decreasing the learning rate, a simple learning rate scheduler was created that allowed the model to fine-tune its parameters and minimize loss as much as possible.

The model was trained for 3 epochs with learning rates of .1, .05, .01, .005, and .001, resulting with a final MSE loss of 0.75. In other words, all predictions made for a user-artist pair had an average error of about 0.86. Given that all ratings in the training and testing datasets are within the range [0, ~60], an average error of .86 is relatively low, which hints that the model fit the data relatively well!

VI. EXPERIMENTAL RESULTS

Figure [5] and Table [2] shows the RMSE on test set for different model using 80% and 20% of the ratings. The main observations are as follows:

1. All models achieve lower RMSE score using more ratings for training. The smallest RMSE score with 20% data (1.224) is still higher than the largest RMSE score with 80% data (1.139). This shows that if possible, having more training data is more important than picking the best model.

2. The effect of constraining the user latent matrix in the cPMF model is significant. Even without utilizing any side information, it achieves comparable performance with the kernelized models that do utilize extra side information.

When the training data is extremely sparse, it provides over 40% reduction in RMSE compared to the PMF baseline model. Since a lot of the times, side information is not readily available, cPMF can be extremely useful in those settings.

3. Exploiting user and artist side information are both effective at reducing RMSE. However, the effect of including artist tag assignment is much more significant than user network. The ineffectiveness of exploiting user network could be due to the fact that user interactions on last.fm website is not a major feature that's actively used by its users. So the network data may be very noisy.

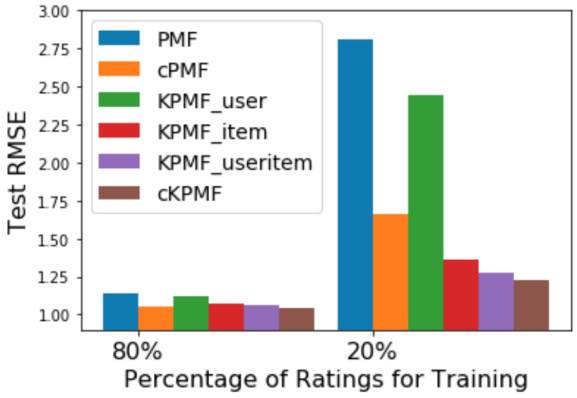


Figure 5. Test RMSE for different models

MODEL	80%TRAIN	20%TRAIN
PMF	1.139	2.808
CPMF	1.056	1.662
KPMF USER	1.119	2.444
KPMF ITEM	1.070	1.361
KPMF USERITEM	1.064	1.273
CKPMF	1:039	1:224

Table 2. RMSE comparision on test set. Smaller is better.

4. Our novel cKPMF model outperforms all other models in both training data settings, with improvement in RMSE more significant when the training data is sparse.

To observe the models' performances for infrequent users, we grouped the users by their number of observed ratings.

We then plot the percentage of improvement in RMSE over the baseline PMF for the various user groups when trained under 80% ratings (Figure [6]).

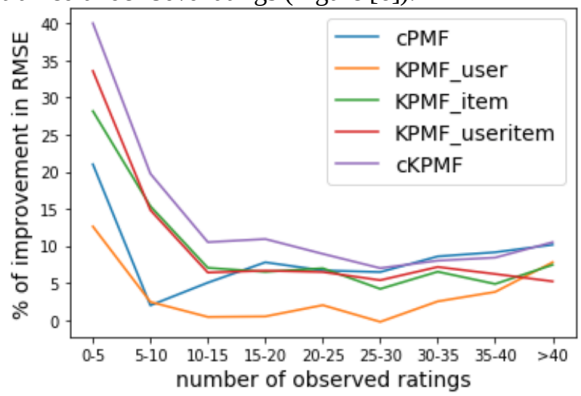


Figure 6. Percentage of improvement in RMSE over PMF by user group

The main observations are as follows:

1. All models achieves higher percentage of improvements for users with very few observed ratings (0 to 5). This is very promising, as we developed these model variations on PMF specifically to address the situations when ratings data are sparse. cPMF's ability to generalize for users with few ratings is especially impressive since it is not utilizing any side information at all.

2. Our cKPMF model outperforms all other models in nearly all user groups (only slightly exceeded by cPMF for users with large enough number of ratings).

3. Comparing performance of cPMF and cKPMF, we see that, when the users rating data is lacking, cKPMF is able to leverage the artist tag information to achieve a boost in performance. But when the rating data is abundant, cPMF and cKPMF achieve similar RMSE.

4. Comparing performance of KPMF item and KPMF useritem, we once again observe that, the effect of incorporating user network information is minimal. But there is still an observable effect for users with nearly no ratings at all.

VII. CONCLUSION

In this project, several existing variations of PMF models is applied and cKPMF model is proposed by combining the techniques in these variations. We explored different kernel functions to best incorporate side information into the KPMF models. cPMF model is constrained by the user latent matrix with a latent similarity matrix, and it is extremely effective at enhancing model performance when side information is unavailable, even for sparse training data.
Leveraging artist tag assignment is much more useful than leveraging user network information for KPMF model. It is observed that our novel cKPMF model is superior to all other models under both training size settings and among all user groups.

For future work, we want to explore the effect of adding user and artist bias into all these models. We want to add memory based models that uses only the side information as baselines. We would also like to experiment with the computational efficiency and convergence behavior of these models in different settings.

VIII. REFERENCES

[1] Little R.J.A., and Rubin D.B. (1987). Statistical Analysis with Missing Data, John Wiley & Sons, New York.

[2] Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen (1984). Classification and regression trees, CRC press.

[3] Kiri Wagstaff., (2004). Clustering with missing values: No imputation required. In Classification, Clustering, and Data Mining Applications, (pp.649-658), Springer.

[4] Ekstrand, M. D., Riedl, J. T., Konstan, J. A., et al. (2011). "Collaborative filtering recommender systems. Foundations and Trends in Human–Computer Interaction", 4(2):81–173.

[5] Salakhutdinov, R. and Mnih, A. (2008). Bayesian probabilistic matrix factorization using markov chain monte carlo", In Proceedings of the 25th international conference on Machine learning, (pp. 880–887), ACM.

[6] Zhou, T., Shan, H., Banerjee, A., and Sapiro, G., Kernelized probabilistic matrix factorization: Exploiting graphs and side information, In Proceedings of the 2012 SIAM international Conference on Data mining, (pp. 403–414).

[7] Fang, H., Bao, Y., and Zhang, J. (2014). Leveraging decomposed trust in probabilistic matrix factorization for effective recommendation, In Twenty -Eighth AAAI Conference on Artificial Intelligence.

[8] L. Gruenwald and M. Halatchev. (2005). Estimating Missing Values In Related Sensor Data Streams, The 11th International Conference Management of Data (COMADO5), (pp. 83-94).

[9] Li Gruenwald, N. Jiang and L. Gruenwald. (2007). Estimating Missing Data In Data Streams, 12th International Conference on Database Systems for Advanced Applications, Bangkok, Thailand, (pp. 981-987).

[10] Y.Li and L. E. Parker. 2008. "A Spatial-Temporal Imputation Technique For Classification With Missing Data In A Wireless Sensor Network", IEEE International Conference on Intelligent Robots and Systems, Nice, France, September 22-26.

[11] Linghe Kong, Mingyuan Xia, Xiao-Yang Liu, Guangshuo Chen, Yu Gu, Min-You, Wuand Xue Liu. (2014). Data Loss and Reconstruction In Wireless Sensor Networks, IEEE transactions on parallel and distributed systems, VOL. 25, NO. 11.

[12] Berihun Fekade, Taras Maksymyuk, Maryan Kyryk, and Minho Jo. (2017). Probabilistic Recovery of Incomplete Sensed Data in IoT, IEEE.

[13] Dimitris Bertsimas, Colin Pawlowski , Ying Daisy Zhuo. (2018). From Predictive Methods To Missing Data Imputation: An Optimization Approach, Journal of Machine Learning Research 18.