# Mining Concepts from Incomplete Datasets Utilizing Matrix Factorization

Lenka Pisková[1], Štefan Pero[1], Tomáš Horváth[2], Stanislav Krajči[1]

[1] ICS, Pavol Jozef Šafárik University, Košice, Slovakia
{Lenka.Piskova, Stefan.Pero}@student.upjs.sk
Stanislav.Krajci@upjs.sk
[2] ISMLL, University of Hildesheim, Germany
horvath@ismll.de

**Abstract.** Formal Concept Analysis aims at finding clusters (concepts) with given properties in data. Most techniques of concept analysis require a dense matrix with no missing values on the input. However, real data are often incomplete or inaccurate due to the noise or other unforeseen reasons. This paper focuses on using matrix factorization methods to complete the missing values in the input data such that it can be used with arbitrary concept analysis technique. The used matrix factorization model approximates the sparse object-item data matrix by a product of two dense factor matrices, thus, mapping objects and items to a common latent space. The mentioned object-factor and item-factor matrices are obtained by a simple stochastic gradient optimization method. We also investigate how the amount of missing values influences the output of the concept analysis. Two measures, well-known in the information retrieval community, have been used for the evaluation of the proposed framework. Real datasets from the UCI Machine Learning Repository were used in our experiments.

**Keywords:** Formal Concept Analysis, matrix factorization, missing values completion, clustering

## 1 Introduction

Formal Concept Analysis (FCA) deals with data in the form of a table which rows represent objects and columns represent attributes of objects. A table entry corresponding to an object $x$ and an attribute $y$ indicates whether or not the object $x$ has the attribute $y$. The clusters on the output of FCA are called concepts, each of which consists of a set of formal objects and a set of formal attributes, called the extent and the intent, respectively, of the given concept. The set of all concepts ordered by $\leq$ forms a concept lattice [1]. The data table required on the input of FCA has to contain all information, i.e. should not contain missing values.

However, real datasets are often incomplete or inaccurate due to the damage or inconsistency in the data collection process. Incomplete context was first


ISBN 978–84–695–5252–0, Universidad de Málaga (Dept. Matemática Aplicada)

introduced by Burmeister and Holzer [2] and applied to attribute implications and attribute exploration. The analysis of incomplete data in the field of FCA can be accomplished in one of the following two ways: The first one is to adapt concept techniques to cope with missing values. Some research was conducted, recently, to generate a concept lattice of an incomplete context [7]. The approach presented in [8] is based on many-valued contexts and conceptual scaling. The second way of handling incomplete data in FCA is to complete the missing values in a pre-processing step and use a concept analysis on the pre-processed data.

The contribution of this paper follows the latter case described above, namely, to predict the missing values using a matrix factorization method (MF) in the pre-processing step such that FCA can be used on the completed data. MF, one of the useful techniques in data mining, allows the decomposition of a sparse[1] matrix into two dense matrices such that the product of these two dense matrices results in a dense matrix which is an approximation[2] of the original, sparse matrix. We use stochastic gradient descent matrix factorization method (SGD MF) in this paper, which is fast and effective. It is important to note that we need to adjust the predicted values since these should be from the input values. The main reason of it is that the resulting dense matrix is just an approximation of the original one with specific values, e.g. 0 and 1 or 1, 2, 3, 4, 5, etc. The resulting dense matrix is then scaled [1] and feed to FCA.

We also address the problem of the robustness of formal concepts, i.e. the ability to remain unaffected by small variations in an input table. As far as we know, there is no related work investigating the issue that how the concepts computed from a full table differs from those computed from a pre-processed table, i.e. what happens when we remove some values from the original table, complete them with a pre-processing method and compute the concepts on the pre-processed table? We think that this question is quite important for a real-life application of FCA.

## 2 Related Work

One direction of estimating missing values in an input matrix is by the use of association rule mining techniques consisting of two approaches: The first one discards instances which contain missing data and generate association rules from the remaining complete data. However, excluding instances with missing data can bias the result. The second approach takes into account the presence of missing values in the rule mining process. Note that two or more association rules may result to different predicted values, thus the so called conflict problem have to be tackled here. The large number of association rules and the effort to reduce the conflict problem have led to the usage of generic bases of association rules. We refer to [3], [4] for details and further references. The percentage of correctly completed missing values is affected by the number of missing values.

[1] We will call a matrix with missing values sparse.

[2] The difference of the values in the non-empty cells of the original matrix and the predicted values for these cells is minimal.

A "hidden" problem is the percentage of missing values that these approaches permits to complete.

[5] and [6] show how to use extracted knowledge represented by formal concepts for completing a relation or at least augmenting it in the case when we do not get each missing value. They generate concepts from the sparse matrix such that they remove rows (objects) containing missing values. However, these approaches have two drawbacks, namely, that the large number of concepts (also for a small relation) makes more difficult to predict missing values, and, more missing values leads to more biased completion.

Combinations of MF and FCA are introduced in [9], [10] and [11], where singular value decomposition, semi-discrete decomposition and non-negative matrix factorization are used for reducing the number of formal concepts. These works, however, use matrix factorization for a different purpose (reduction of the number of concepts) and consider full input matrix.

A novel approach to combine matrix decomposition and factor analysis is presented in [12], [13]. There are two main differences from ordinary factor analysis. First, a table entry represents a degree to which an object has an attribute (table entries are from a bounded scale $L$). Second, the matrix composition operator is a so-called t-norm-based product. The aim of this approach is not the completion of missing values but finding a set of factors that correspond to formal concepts.

## 3 Preliminaries

We will just briefly describe FCA, and focus instead on a more detailed description of the used MF technique in this section since we think it could be more interesting to the FCA community.

### 3.1 Formal Concept Analysis

A formal context is a triple $(X, Y, I)$ consisting of two non-empty sets $X$ and $Y$ and a binary relation $I$ between them. The elements of $X$ are called the objects and the elements of $Y$ are called the attributes. $(x, y) \in I$ means that the object $x$ has the attribute $y$.

For a set $A \subseteq X$ and a set $B \subseteq Y$, define $A' = \{y \in Y : (\forall x \in A)(x, y) \in I\}$ and $B' = \{x \in X : (\forall y \in B)(x, y) \in I\}$. $A'$ is the set of attributes common to the objects in $A$ and $B'$ is the set of object which have all attributes in $B$.

A formal concept of the context $(X, Y, I)$ is a pair $(A, B)$ of a set $A \subseteq X$ of objects and a set $B \subseteq Y$ of attributes such that $A' = B$ and $B' = A$. $A$ and $B$ are called the extent and the intent of the concept $(A, B)$, respectively. Denote the set of all concepts in $(X, Y, I)$ by $\mathcal{B}(X, Y, I)$.

Introduce a relation $\leq$ on $\mathcal{B}(X, Y, I)$ by $(A_1, B_1) \leq (A_2, B_2)) \Leftrightarrow A_1 \subseteq A_2 \Leftrightarrow B_2 \subseteq B_1$, $(A_1, B_1)$ is called the subconcept of $(A_2, B_2)$ and $(A_2, B_2)$ is called the superconcept of $(A_1, B_1)$. The set of concepts $\mathcal{B}(X, Y, I)$ ordered by $\leq$ constitutes the concept lattice (or Galois lattice) of the context $(X, Y, I)$. The so-called main

theorem of concept lattices characterizes the structure of concept lattices. For further details we refer to [1].

### 3.2 Stochastic Gradient Descent Matrix Factorization

Stochastic gradient descent matrix factorization (SGD MF) is one of the most popular factorization techniques [14] because of its scalability and good accuracy.

The goal of this approach is to approximate a sparse matrix $X \in \mathbb{R}^{|\mathcal{O}|\times|\mathcal{I}|}$ with a product of two (much smaller) matrices $W \in \mathbb{R}^{|\mathcal{O}|\times K}$ and $H \in \mathbb{R}^{|\mathcal{I}|\times K}$ such that

$$X \approx \hat{X} = WH^T, \tag{1}$$

where $|\mathcal{O}|, |\mathcal{I}|$ are the number of objects and items, respectively, and $K$ is number of (latent) factors. The $i$th row of the matrix $W$ is a vector containing $K$ latent factors describing the object $i$, and the $j$th row in matrix $H$ is a vector containing $K$ latent factors describing the item $j$ [15].

The estimate $\hat{x}_{ij}$ of a missing value at the row $i$ and column $j$ of the sparse matrix $X$ is computed as

$$\hat{x}_{ij} = (WH^T)_{ij} = \sum_{k=1}^{K} w_{ik}h_{jk} \tag{2}$$

where $w_{ik}$ ($h_{jk}$) is the value of $W$ ($H$) at the $i$th ($j$th) row and $k$th column.

We are interested in such $\hat{X}$ which estimates the missing values of $X$ well. Since we do not know the missing values, we use the following trick, well-known in the machine learning community [18]: We split $X$ into two complementary parts, called the train set $X^{train}$ and the test set $X^{test}$. The model $\hat{X}$ is then approximated from $X^{train}$ and it's quality is assessed by the root mean squared error (RMSE) loss computed on $X^{test}$, defined as

$$RMSE = \sqrt{\frac{\sum\limits_{x_{ij}\in X^{test}} (x_{ij} - \hat{x}_{ij})^2}{|X^{test}|}}, \tag{3}$$

where $x_{ij}$ are the values of the non-empty cells of $X$ belonging to $X^{test}$. In this way, we simulate the case when we have a sparse matrix ($X^{train}$) which missing values ($X^{test}$) are known, however.

Our goal is to find those parameters $W$ and $H$ of the model $\hat{X}$ for which the RMSE is minimal. However, we have to keep in mind that $X^{test}$ is "hidden", i.e. it represents the "unknown" missing values, thus, we have to use[3] the data we know, i.e. the train set $X^{train}$. Since $|X^{train}|$ is constant, minimizing RMSE (3) on $X^{train}$ is equivalent to minimizing the sum of errors $\sum_{x_{ij}\in X^{train}}(x_{ij} - \hat{x}_{ij})^2$. Since RMSE is a (loss) function, we use a stochastic gradient descent (SGD)

[3] Here we also expect that $X^{test}$ has similar statistical characteristics as $X^{train}$.

optimization method [16] for searching those parameters $W$ and $H$ of RMSE such that the following objective function is minimal:

$$\sum_{x_{ij} \in X^{train}} (x_{ij} - \hat{x}_{ij})^2 + \lambda(\|W\|^2 + \|H\|^2) \tag{4}$$

where $\hat{x}_{ij}$ is defined as in (2) and $\lambda$ is a regularization term to prevent the so-called overfitting (i.e. when a model estimates very well the training data but poorly the test data [18]). $\lambda$ controls the magnitudes of the factor vectors such that $W$ and $H$ would give a good approximation of the whole original input matrix $X$ (containing both $X^{train}$ and $X^{test}$). $\|W\|^2$ means square of the one vector in matrix W.

In the training phase we first initialize two matrices $W$ and $H$ with some random values (for example from the normal distribution $N(0, \sigma^2)$ with mean 0 and standard deviation 0.01) and compute the estimation error

$$err = \sum_{x_{ij} \in X^{train}} e_{ij}^2 \tag{5}$$

where

$$e_{ij}^2 = (x_{ij} - \hat{x}_{ij})^2 = (x_{ij} - \sum_{k=1}^{K} w_{ik} h_{jk})^2 \tag{6}$$

We minimize this error by updating the values of $W$ and $H$ in the following way [16]: we randomly choose a value $x_{ij}$ from $X^{train}$ and compute the gradient of the objective function (6) in this value w.r.t. the parameters[4] $w_i$ and $h_j$, i.e.

$$\frac{\delta}{\delta_{w_{ik}}} e_{ij}^2 = -2 e_{ij} h_{jk} = -2(x_{ij} - \hat{x}_{ij}) h_{jk} \tag{7}$$

$$\frac{\delta}{\delta_{h_{jk}}} e_{ij}^2 = -2 e_{ij} w_{ik} = -2(x_{ij} - \hat{x}_{ij}) w_{ik} \tag{8}$$

In the next step we use the computed gradient to update the values of $w_{ik}$ and $h_{jk}$:

$$w_{ik}^{(new)} = w_{ik} - \beta \frac{\delta}{\delta_{w_{ik}}} e_{ij}^2 = w_{ik} + 2\beta e_{ij} h_{jk} \tag{9}$$

$$h_{jk}^{(new)} = h_{jk} - \beta \frac{\delta}{\delta_{h_{jk}}} e_{ij}^2 = h_{jk} + 2\beta e_{ij} w_{ik} \tag{10}$$

where $\beta$ is the learning rate controlling the step sizes.

We update $W$ and $H$ for each value of $X^{train}$ in one iteration. The number of iterations, i.e. the number of passes over the full $X^{train}$ is a hyper-parameter of the factorization technique as well as the regularization term $\lambda$, the learn rate $\beta$ and the number of factors $K$.

[4] Only the vectors $w_i$ and $h_j$ contribute to $e_{ij}^2$, thus we treat the other vectors (rows of $W$ and $H$) as constants (which derivatives are zero).

As we see, the quality of the model $\hat{X}$ depends on the hyper-parameters used in the learning phase. Also, it can happen that the split of the data into the train and the test part was done in a way that these two parts have no similar characteristics, e.g. one part contains large values while the other the low ones. To prevent this phenomenon to happen we usually use $n$-fold cross validation [18] in which we split $X$ into $n$ equal parts $X_1, \ldots, X_n$ and for each $n$ and hyper-parameter combination $(K, \lambda, \beta, iterations)$ do the following: train a model $\hat{X}^n$ with the given hyper-parameter combination on $\cup_{i \neq n} X_i$ and compute its RMSE on $X_n$. The resulting RMSE of $\hat{X}$ with the hyper-parameters $(K, \lambda, \beta, iterations)$ is then the average RMSE over all $\hat{X}^n$. After trying some hyper-parameter combinations with $n$-fold cross validation, we choose the best combination $(K, \lambda, \beta, iterations)_{best}$ which has the smallest average RMSE over all folds. Finally, we train the model $\hat{X}$ again using the whole input matrix $X$.

It can happen, that the input matrix $X$ contains an empty line or column with no values. For those objects (rows) or items (columns) we will not update the parameters $W$ and $H$ (it follows from the algorithm of SGD MF). In such cases, estimated missing values could be the global average of non-missing values. More advanced and sophisticated methods that can improve the estimation in such cases are listed in [17].

## 4 Illustrative example

To illustrate our approach we present a small toy example. Our purpose is to cluster students into groups with respect to their success in solving the tasks. The key problem is that not every student have solved all of the tasks. We will proceed as follows. We predict the success of students in tasks that they have not solved yet (empty values in the input table). After the process of conceptual scaling is completed, we find formal concepts.

The table in the left side of the figure 1 contains five students who have performed several tasks. The total marks that can be scored are 9; 3 marks for each task. Table values represent students' marks for tasks. The matrix represents a relation between objects and items (which are students and tasks in our case).

Figure 1 shows an example of how we can factorize the students and tasks. After the training phase with $K = 2$ latent factors (F1 and F2), we get the student-factor matrix $W$ and the factor-task matrix $H$. Suppose we would like to predict Tom's (3rd row) performance for the task 2 (2nd column). The prediction is performed using equation 2

$$\hat{x}_{32} = \sum_{k=1}^{K} w_{3k} h_{2k} = 1.80 * 1.20 + 0.13 * 1.08 = 2.30.$$

Since the predicted value should be included in the input matrix, we round this value to one of the values 1, 2 and 3. We have predicted Tom will achieve average results in task 2. Similarly, we can predict the performance of other

| X | T1 | **T2** | T3 |
|---|---|---|---|
| Katie | 3 | 3 | |
| Dean | 2 | 2 | 1 |
| **Tom** | 2 | | 2 |
| Sam | | 2 | |
| Adam | 2 | 3 | 2 |

| W | F1 | F2 |
|---|---|---|
| Katie | 1.18 | 1.52 |
| Dean | 1.24 | 0.31 |
| Tom | 1.80 | 0.13 |
| Sam | 0.91 | 1.57 |
| Adam | 1.68 | 0.49 |

| H | T1 | T2 | T3 |
|---|---|---|---|
| F1 | 0.98 | 1.20 | 0.85 |
| F2 | 1.35 | 1.08 | 0.54 |

**Fig. 1.** Factorization of the input matrix.

students in tasks which they have not done yet. The full matrix is depicted in the left side of the figure 2.

Now, we would like to cluster students according to their results in solving the tasks. The table in the left side of the figure 2 represents many-valued context. Since values are ordered and each value implies the weaker one, we use the ordinal scale in the right side of the figure 2. Performing a task at the highest level implies performing the task at the average level, too. The table in the figure 3 is the result of the conceptual scaling process. For details on many-valued contexts and conceptual scaling we refer to [1].

| | T1 | T2 | T3 |
|---|---|---|---|
| Katie | 3 | 3 | 2 |
| Dean | 2 | 2 | 1 |
| Tom | 2 | 2 | 2 |
| Sam | 3 | 2 | 2 |
| Adam | 2 | 3 | 2 |

| | 3 | 2 | 1 |
|---|---|---|---|
| 3 | × | × | × |
| 2 | | × | × |
| 1 | | | × |

**Fig. 2.** The many-valued context and the ordinal scale $S$ for all attributes of the many-valued context.

The obtained concept lattice contains 6 concepts (clusters) and is depicted in the figure 3. A labeling is simplified by putting down each object and attribute only once. The extent of each concept is formed by collecting all objects which can be reached by descending line paths from the concept. The intent of each concept consists of all attributes located along ascending line paths.

We interpret some of the clusters, here: Each student solved the task 1 and task 2 at least at average level (the top element of the lattice). All students except for Dean achieved at least average results in solving all tasks. Nobody performed good results in every task (the bottom element of the lattice).

## 5 Experiments

Experiments were conducted on computing cluster with 7 computing nodes, each of which has 16 cores, running Red Hat Linux.

| | T1 | | | T2 | | | T3 | | |
|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 1 | 2 | 3 | 1 | 2 | 3 |
| Katie | × | × | × | × | × | × | × | × | |
| Dean | × | × | | × | × | | × | | |
| Tom | × | × | | × | × | | × | × | |
| Sam | × | × | × | × | × | | × | × | |
| Adam | × | × | | × | × | × | × | × | |

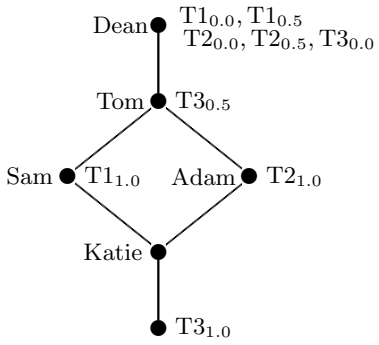


**Fig. 3.** The one-valued context as the derived context of the many-valued context in figure 2 and the corresponding concept lattice.

### 5.1 Data

For these experiments, we consider a complete database to act as a reference database, and we randomly introduce missing values with the following different rates : 10%, 20% and 30%. Benchmark datasets used for this experiments are from the UCI Machine Learning Repository[5]. Characteristics of these datasets are depicted in the table 1.

**Table 1.** Datasets characteristics

| Dataset | Objects | Attributes | Attribute types | FCA attributes |
|---|---|---|---|---|
| Wine | 178 | 13 | Integer, Real | 68 |
| SPECT Heart | 267 | 22 | Categorical | 46 |
| Tic-tac-toe | 958 | 9 | Categorical | 29 |

The data conversion into FCA format was done as follows. Categorical (many-valued) attributes were converted by creating a formal context attribute for each of the values. Attributes with integer or real types were scaled to create context attributes with ranges of values.

### 5.2 Matrix factorization model settings

We implemented matrix factorization model in Java described in the section 3.2 on our own. We have used 3-fold cross-validation for testing the proposed model: In each of the 3 iterations, one fold was used for testing. The remaining two folds were used for tuning hyper-parameters of the model. Hyper-parameters (number

[5] http://www.ics.uci.edu/mlearn/MLRepository.html.

of factors, number of iterations, learn rate and regularization term) were tuned using grid search [19] (a systematic search over different combinations). The best hyper-parameter combinations for our model and the root mean squared error (RMSE) was measured on the test fold for each dataset. The best hyper-parameters are reported in the table 2.

**Table 2.** The best hyper-parameters found by grid search

| Dataset | Missing values (%) | Factors | Iterations | Learn rate | Reg. term |
|---|---|---|---|---|---|
| Wine | 10 | 12 | 30 | 0.055 | 0.1 |
| | 20 | 12 | 40 | 0.055 | 0.05 |
| | 30 | 12 | 50 | 0.055 | 0.05 |
| SPECT Heart | 10 | 22 | 50 | 0.055 | 0.05 |
| | 20 | 42 | 40 | 0.055 | 0.05 |
| | 30 | 72 | 40 | 0.055 | 0.05 |
| Tic-tac-toe | 10 | 42 | 70 | 0.005 | 0.05 |
| | 20 | 12 | 60 | 0.005 | 0.05 |
| | 30 | 32 | 90 | 0.005 | 0.1 |

### 5.3 Results

Concept lattices from complete data sets were computed, the number of concepts is shown in the table 3.

**Table 3.** Datasets and the number of concepts computed from the full matrix.

| | number of concepts |
|---|---|
| Wine | 24423 |
| SPECT Heart | 2135549 |
| Tic-tac-toe | 59505 |

Table 4 presents the comparison of the RMSE measured over 3 folds and average of success rates of the prediction for all datasets. Using matrix factorization model we correctly predicted from 30% to 80% missing values. If we did not predict a missing value correctly then the predicted value was close to the original value.

In order to evaluate methods for completing missing values to mine formal concepts, one have to compare concepts generated from the original data table (initial context) with concepts generated from the data table containing estimated values (predicted context).

**Table 4.** RMSE and the average of success rates of the prediction

| Dataset | Missing values (%) | RMSE | Average of the success rates (%) |
|---|---|---|---|
| Wine | 10 | 0.6944168 | 53.18 |
| | 20 | 0.7345221 | 66,46 |
| | 30 | 0.6972556 | 48,84 |
| SPECT Heart | 10 | 0.37255815 | 78,61 |
| | 20 | 0.3785878 | 78,09 |
| | 30 | 0.38591003 | 77,37 |
| Tic-tac-toe | 10 | 0.88052654 | 31,72 |
| | 20 | 0.9031108 | 31,56 |
| | 30 | 0.9076978 | 33,71 |

We propose to use two measures for evaluating the performance of the proposed approach for the missing values completion with respect to mined concepts. Both are well-known in the information retrieval (IR) community. The first one indicates how many concepts of the initial context occur in the set of concepts of the estimated context. The second one determines how many concepts of the estimated context are in the set of concepts of the initial context.

Let $(X, Y, I)$ be the initial formal context and $(X, Z, J)$ be the estimated formal context. Let $O$ be the set of formal concepts of the initial context and $E$ be the set of formal concepts of the estimated one ($O = \mathcal{B}(X, Y, I)$ and $E = \mathcal{B}(X, Z, J)$). We propose to use the following two measures:

$$precision = \frac{|O \cap E|}{|E|} \text{ and } recall = \frac{|O \cap E|}{|O|}.$$

Precision is the fraction of concepts of the estimated context that are in the set of concepts of the initial context. Recall is the fraction of concepts of the initial context that are in the set of concepts of the estimated context. The higher the precision, the more concepts of the estimated context are in the set of concepts of the initial context. The higher the recall, the more concepts of the initial context are in the set of concepts of the estimated context.

Two formal concepts are equal when any formal object in the extent of one is in the extent of the other and vice versa, the same for formal attributes. In this case, *precision* and *recall* are computed. From the extent-ional point of view: two concepts are equal when they have the same extent. Analogously, from the point of view of intent, two concepts are equal when they have the same intent. In these cases, we calculate $precision_e$, $recall_e$ and $precision_i$, $recall_i$, respectively.

The results of our experiments are shown in the table 4 and 5.

It is noteworthy that there is a huge difference between $precision_e$ and $precision_i$. We assume that the gap between $precision_e$ and $precision_i$ is caused by the fact that the number of objects is greater than the number of attributes.

Moreover, the precision and recall are quite small. The reason is how we have measured the equality of the sets: the sets are not equal if they differ even only in one element. Nevertheless, the proposed measures are useful for experimental

**Table 5.** Precision and recall.

| Dataset | Missing values (%) | $prec$ | $prec_e$ | $prec_i$ | $rec$ | $rec_e$ | $rec_i$ |
|---|---|---|---|---|---|---|---|
| Wine | 10 | 0,0005 | 0,0277 | 0,2603 | 0,0005 | 0,0311 | 0,2920 |
| | 20 | 0,0004 | 0,0246 | 0,2603 | 0,0004 | 0,0272 | 0,2883 |
| | 30 | 0,0002 | 0,0194 | 0,2187 | 0,0002 | 0,0198 | 0,2238 |
| SPECT Heart | 10 | $6 \times 10^{-5}$ | 0,0107 | 0,3775 | $4 \times 10^{-6}$ | 0,0007 | 0,0257 |
| | 20 | $4 \times 10^{-5}$ | 0,0063 | 0,3667 | $3 \times 10^{-6}$ | 0,0005 | 0,0279 |
| | 30 | $2 \times 10^{-5}$ | 0,0039 | 0,3346 | $2 \times 10^{-6}$ | 0,0003 | 0,0267 |
| Tic-tac-toe | 10 | 0,0011 | 0,0239 | 0,3969 | 0,0002 | 0,0032 | 0,0524 |
| | 20 | 0,0011 | 0,0132 | 0,3899 | 0,0001 | 0,0009 | 0,0258 |
| | 30 | 0,0008 | 0,0139 | 0,3477 | 0,0001 | 0,0011 | 0,0282 |

comparisons of various techniques used in the pre-processing step. We will focus on finding other measures. We suppose that if we modify measures to use the similarity (of extents, intents and concepts) better results could be achieved.

## 6 Conclusion and future research directions

In this paper we have introduced framework for mining concepts from incomplete datasets. The proposed framework uses stochastic gradient descent matrix factorization (SGD MF) in the pre-processing step and after completing the missing values a formal concept analysis (FCA) is deployed. Using SGD MF we are able to quite precisely predict values from a sparse data matrix without having any background knowledge about the data. Two measures (precision and recall) have been used for evaluating the presented framework.

The experiments on three datasets (with 10, 20 and 30 percent of missing values) showed that the percentage of missing values does not influence the prediction rate, and, that there is a large difference in the precision and recall measures when computed on the extents and the intents of the resulting concepts, respectively.

Our further research will focus on modifying the (precision and recall) measures with using the similarity of extents, intents and concepts instead of their equality. The generation of all concepts from the huge incidence matrix is time consuming and the number of concepts is very large. Using our matrix factorization model we transform input matrix into more smaller latent factor matrices $W$ and $H$. These matrices describe objects and items according to latent factors. We want to explore these matrices and to use them for reducing the number of concepts.

Even if there are some remaining issues to investigate, experimental results show that the proposed approach is promising and worth of further research.

**Acknowledgements:** This work was supported by the grants VEGA 1/0832/12 and VVGS-PF-2012-22 at the Pavol Jozef Šafárik University in Košice, Slovakia.

## References


1. B. Ganter, R. Wille: Formal concept analysis: Mathematical foundations. Springer, 1999, ISBN 3-540-62771-5.
2. P. Burmeister, R. Holzer: Treating Incomplete Knowledge in Formal Concept Analysis. In: B. Ganter, G. Stumme, R. Wille (Eds.): Formal Concept Analysis: State of the Art, LNAI 3626, Springer, Heidelberg, 2005.
3. L. Ben Othman, S. Ben Yahia: Yet Another Approach for Completing Missing Values. CLA 2006, pages 155-169.
4. L. Ben Othman, S. Ben Yahia: $GBAR_{MVC}$: Generic Basis of Association Rules Based Approach for Missing Values Completion. The International Journal of Computing and Information Sciences (IJCIS), 2008.
5. S. Ben Yahia, Kh. Arour, A. Jaoua: Completing Missing Values Using Discovered Formal Concepts. DEXA 2000, pages 741-751.
6. S. Ben Yahia, Kh. Arour.: Mining fuzzy formal concepts for completing missing values. In Proceedings of the Intl. Symposium on Innovation in Information and Communication Technology, Amman, Jordan, 2002.
7. M. Krupka, J. Lastovicka: Concept lattices of incomplete data. In: F. Domenach, D.I. Ignatov, and J. Poelmans (Eds.): ICFCA 2012, LNAI 7278. Springer, Heidelberg, pages 180-194, 2012.
8. J. Liu, X. Yao: Formal Concept Analysis of Incomplete Information System. In: FSKDIEEE (2010), pages 2016-2020.
9. P. Gajdoš, P. Moravec, V. Snášel: Concept Lattice Generation by Singular Value Decomposition, In: V. Snášel, R. Bělohlávek (Eds.): CLA 2004, pages 102-110.
10. V. Snášel, P. Gajdoš, H. Abdulla, M. Polovinčák: Using Matrix Decompositions in Formal Concept Analysis. ISIM 2007, pages 121-128.
11. V. Snášel, H. Abdulla, M. Polovinčák: Using Nonnegative Matrix Factorization and Concept Lattice Reduction to visualizing data. SYRCoDIS, 2007.
12. R. Bělohlávek, V. Vychodil: Factor Analysis of incidence data via Novel Decomposition of Matrices, In: S. Ferré and S. Rudolph (Eds.): ICFCA 2009, LNAI 5548, pages 83-97, 2009.
13. R. Bělohlávek, V. Vychodil: Discovery of Optimal Factors in binary data via novel method of matrix decomposition. Journal of Computer and System Sciences 76(1)(2010), pages 3-20.
14. Y. Koren, R. Bell and C. Volinsky. Matrix Factorization Techniques for Recommender Systems. COMPUTER (Vol. 42, 8): IEEE Computer Society Press, Los Alamitos, USA, 2009.
15. G. Takács, I. Pilászy, B. Németh. Investigation of Various Matrix Factorization Methods for Large Recommender Systems. KDD Netflix workshop, 2008.
16. L. Bottou. Stochastic learning. In Bousquet, O. and von Luxburg, U., editors, Advanced Lectures on Machine Learning, Lecture Notes in Artificial Intelligence, LNAI 3176, pages 146-168. Springer Verlag, Berlin.
17. Z. Gantner, L. Drumond, C. Freudenthaler, S. Rendle, L. Schmidt-Thieme. Learning attribute-to-feature mappings for cold-start recommendations. In Proceedings of the 10th IEEE International Conference on Data Mining (ICDM 2010). IEEE Computer Society, 2010.
18. T. Hastie, R. Tibshirani, J. Friedman: The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Second Edition, Springer Series in Statistics, 2009.
19. J. Bergstra, Y. Bengio: Random Search for Hyper-Parameter Optimization. J. Machine Learning Research 13, pages 281-305, 2012.