Mining Data Quality Rules for Data Migrations: A Case Study on Material Master Data

Marcel Altendeitering(✉)

Fraunhofer ISST, Emil-Figge-Straße 91, 44227 Dortmund, Germany
marcel.altendeitering@isst.fraunhofer.de

Abstract. Master data sets are an important asset for organizations and their quality must be high to ensure organizational success. At the same time, data migrations are complex projects and they often result in impaired data sets of lower quality. In particular, data quality issues that involve multiple attributes are difficult to identify and can only be resolved with manual data quality checks. In this paper, we are investigating a real-world migration of material master data. Our goal is to ensure data quality by mining the target data set for data quality rules. In a data migration, incoming data sets must comply with these rules to be migrated. For generating data quality rules, we used a SVM for rules at a schema level and Association Rule Learning for rules at the instance level. We found that both methods produce valuable rules and are suitable for ensuring quality in data migrations. As an ensemble, the two methods are adequate to manage common real-world data characteristics such as sparsity or mixed values.

Keywords: Master data · Data quality · SVM · Association rule learning · Data migration

1 Introduction

Data migrations are understood as the process of permanently moving data from a source system to a target system in the right quality [17]. For instance, whenever a new software is introduced or the corporate structure changes (e.g. due to M&A) the need for a data migration arises. Despite the fact that companies are regularly confronted with data migrations their success rates are low. They are often underestimated in size and complexity and companies lack the necessary specialist skills. As a result, the quality of the target data set is impaired by introducing low quality data, which can jeopardize organizational success [16,17].

As data, and in particular master data, is a valuable asset for organizations, it is vital to ensure high quality data. For this, numerous methods and tools are available that can support practitioners in detecting errors in a single column [5]. These are often embedded in database management systems (DBMS) and automatically detect errors like missing values, duplicate entries or invalid categorical data. Detecting errors that involve multiple columns (e.g. functional

T. Margaria and B. Steffen (Eds.): ISoLA 2021, LNCS 13036, pp. 178–191, 2021.
https://doi.org/10.1007/978-3-030-89159-6_12

dependencies) is much more difficult and rarely included in established automated data quality tools [10]. A common approach for identifying multi-column data errors is the 'consulting approach', in which internal or external domain experts clean data sets using data quality rules and metrics [23]. Therefore, the domain experts manually analyze dependencies between values and attributes in the data set and create rules. The downside of this approach is that the need for experts makes the process time-consuming and expensive [9]. Moreover, the experts can miss rules, which are hidden in the data set but not made explicit using quality rules. A solely consulting based approach is therefore not suitable for large scale data migrations, in which many entries are changed at once. There is a need for an automated detection of multi-column data errors that can support data migrations and reduce the amount of data quality work.

In this paper, we propose an extended data migration process that uses data quality rule mining for automated quality checks in data migrations. With data quality rule mining we are uncovering relationships between attributes in relational data sets and are evaluating incoming data against the derived rules. This way, we are able to reduce the amount of data quality work and prevent complex data errors that would normally require expert support. In contrast to other solutions, our work focuses on multi-column errors and uses a combined approach to identify rules at the schema and the instance level. This way we are able to handle diverse data sets and can support the generality of our solution. We conducted and evaluated our study in a real-world migration of a material master data.

The remainder of this paper is structured as follows. In Sect. 2, we describe the experimental setting we investigated in our study. In Sect. 3, we propose the extended data migration process and show how we automatically derive data quality rules to support data migrations. Afterwards, we present the findings of our study in Sect. 4. Finally, in Sect. 5, we will discuss related work and draw a conclusion in Sect. 6.

2 Case Description

2.1 Setting

For our case study, we investigated the data migration process at a large German pharmaceutical company, which we call PharmCo in this study. PharmCo has several affiliated companies and production sites around the globe. The diversified and international nature of the company led to a complex IT and data infrastructure. In order to harmonize the data infrastructure, data migrations are required in regular intervals, causing a substantial financial and organizational effort.

In particular, we investigated the migration of material master data from external sources (e.g. a remote production site) to a central SAP system. The data sets in this system are of high importance for the company and inherit a great value for business operations. For example, incorrect tax information about a material could cause fines and interrupt the supply and delivery chains. Thus, it is important that the overall quality of this database is high and must not be deteriorated during data migrations.

2.2 Current Data Migration Process

Currently, data quality is maintained by an established data migration tool, which automatically enforces data models and secures data quality. However, the data migration system is not capable of detecting complex data quality issues involving multiple attributes, which causes the introduction of new errors to the data set during each migration. The correction of these errors costs PharmCo up to 10,000 Euros per data migration. Specifically, the current data migration process consists of three steps that are conducted in sequence (see Fig. 1).

- *Step 1*: Once the data migration starts the external data sets are imported into the data migration tool. The tool performs schema matching and error detection to harmonize the data sets and find simple data errors.
- *Step 2*: All data that passes the checks in the data migration tool is introduced to the target data set.
- *Step 3*: After the migration is completed there often remain errors in the target data set that were not detected by the data migration tool. Therefore, an experienced employee checks all entries manually to find and resolve potential issues. As this process is very expensive the expert focuses on error prone attributes, that had errors in previous migrations.

To address the limitations of the current approach PharmCo is looking for an automated solution that improves the error detection accuracy of the data migration tool and supports domain experts in resolving errors. During a workshop with two members of the data migration team we were able to derive the general requirements for such a solution. Most importantly, both participants from PharmCo mentioned that errors in a data migration mostly occur on the instance and the schema level. They therefore formulated the requirement that:

> *"The tool needs to generate rules for both, the instance and the schema level, so we can evaluate incoming data against these rules"*

They provided two examples, one for each category respectively. An instance rule is for example one, which specifies that when there is a certain value for the field 'Base Unit' there must also be certain values in the fields 'Material Group' and 'Volume'. A rule on the schema level should for example specify that once the attribute 'Gross weight' is filled with any value the attribute 'Net weight' also needs to be filled. They also specified that these are simple examples and potential rules can include several attributes, which makes the detection and formulation difficult. Another important aspect is the execution time of the tool. One participant mentioned:

> *"Data migrations are time-critical projects. It is therefore important for the algorithms to run in a limited amount of time and be highly scalable."*

Moreover, PharmCo stated that the quality rules the tool derives must be 'actionable'. This means the output should be easy to interpret semantically and enable an integration with existing tools using standardized interfaces.

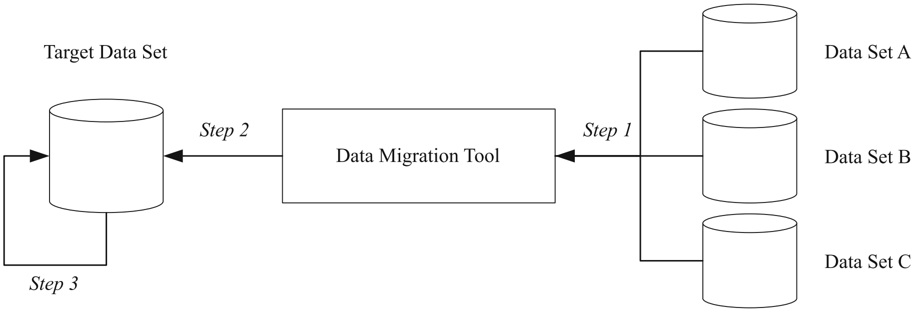

Fig. 1. Current approach for ensuring data quality in data migrations.

2.3 Data Set Description

For our study, PharmCo provided us with access to two real-world material data sets: trading goods and spare parts. The trading goods data set contains information about finished products that are ready to be sold on the market. Among others, this data set provides a description of the good, tax information, relations to other goods. The spare parts data set contains information about materials that are part of trading goods. This data set includes for instance the dimensions of the good, safety restrictions and weights.

There are several reasons why we selected to use material data sets and in particular these two kinds. (1) The material data sets do not contain any personal information and are not regulated by data privacy laws. It was therefore easier to gain access to these data sets as compared to supplier or customer data. (2) The data on trading goods and spare parts do not contain information about specific pharmaceuticals or ingredients, which could reveal company secrets. (3) The data definitions and data schemes vary between different parts of the company, which can lead to a large number of errors in data migrations. (4) The master data sets are important for business operations and can help to raise awareness about data quality.

Since we are working with real-world data sets, we are not able to present the actual data we used to protect intellectual property. However, we are able to describe the schema of the data in more detail. The trading goods data set has 173 attributes (46 constant, 84 categorical, 23 free text, 18 numerical, 2 date) and 15,904 entries. The spare parts data set also features 173 attributes (111 constant, 37 categorical, 7 free text, 12 numerical, 6 date) and has 92,869 entries. Overall, the trading goods data set has a sparsity of 10.5% and a size of 14.8 MB. The spare parts data set has a sparsity of 61.3%, which results in a size of 64.4 MB. The selected data sets originate from a real-world database and feature some typical challenges like high dimensionality, type mix, special characters and sparsity. It was therefore necessary to pre-process the data for analysis. Suitable methods for mining data quality rules must be able to incorporate these characteristics.

2.4 Data Pre-processing

The data sets we obtained were already pre-processed with simple data transformations and quality enforcements (e.g. type enforcement). Such tasks are usually conducted by the Master Data Management (MDM) system and offered us the possibility to focus on more complex errors.

However, the data sets needed further pre-processing to enable the analysis and ensure efficiency. We started by removing all columns with constant or no values as suggested by [15]. Constant values do not provide any value to the machine learning algorithms and improve the performance of our analysis by limiting the dimensionality of the data sets. As a result, we reduced the trading goods data set to 127 attributes and the spare parts data set to 62 attributes. We furthermore removed certain special characters like commas and semicolons from the free text values, which were hindering the import of the csv source files.

3 Proposed Solution

3.1 Extended Data Migration Process

With our study we want to support the currently manual process for ensuring data quality in data migrations with machine learning (ML) techniques. Therefore, we extended the manual process with a new capability that automatically derives data quality rules from a target data set and applies these to the data sets to be migrated. Following Chiang and Miller, data quality rules "define relationships among a restricted set of attribute values that are expected to be true under a given context" (p. 1166) [8]. Usually, these rules are developed manually by domain and business experts. However, obtaining a complete set of data quality rules is difficult as the process is time-consuming and costly. Additionally, there might be rules present in the data set that the experts are not aware of [8].

The proposed 'Data Quality Rule Mining' component tries to automatically identify such rules and apply them in a data migration. This way we want to reduce the amount of manual data quality checks by providing a hint on potential errors, which leads to reduced cost and better data quality. By extending the current data migration process with the proposed component, we get a new process consisting of four steps (see Fig. 2):

– *Step 1*: Using data mining and profiling techniques we derive suitable rules from a target data set on a schema and an instance level. Hereby, we assume that the given data set is correct. Optionally, a human-in-the-loop manually evaluates the derived rules to improve the accuracy of the result.
– *Step 2*: The derived data migration rules are implemented as executable rules in a data migration tool.
– *Step 3*: During a data migration the migration tool automatically checks if incoming data satisfies the rules. In case a check fails the issue is resolved manually or automatically by the tool.
– *Step 4*: Data that passes all checks is migrated to the target data set without further checks.

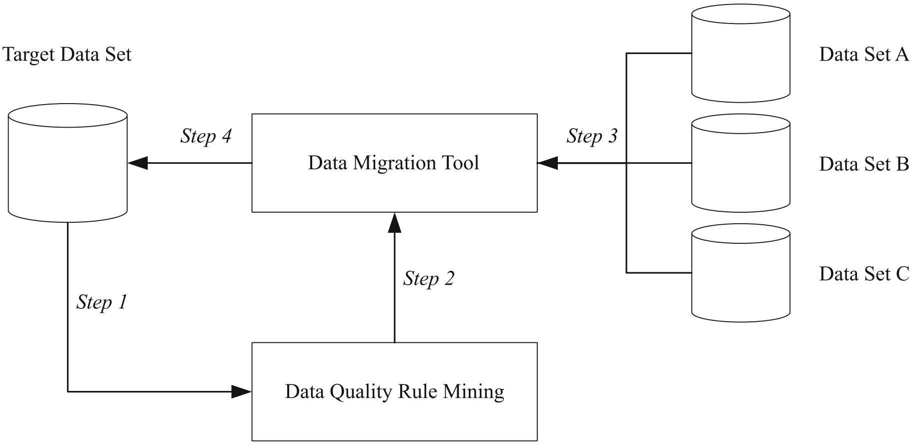

Fig. 2. The role of mining data quality rules in the data migration process.

3.2 Data Quality Rule Mining

Initial Selection. Since there is a large body of research available on mining and profiling relational data sets, we decided to return to the literature for selecting suitable methods for data quality rule generation. Several survey and overview papers in the domains of data mining and data profiling discuss potential approaches [1,2,8,10,12,19]. Following the requirement of PharmCo we can distinguish these methods between the schema and instance levels [19] and the dimensionality of a potential rule [2]. In terms of dimensionality, Abedjan et al. separate data profiling tasks in single-column, multi-column and dependency tasks [2]. In our case, we are only interested in multi-column and dependencies.

In each of these methodological categories there are numerous algorithms available. However, finding and selecting an algorithm that works on real-world data sets and satisfies our requirements remains difficult. The accuracy and usefulness of many methods is unclear when they are applied to a real-world data set. Most of them have either only been tested on synthetic data or on synthetically injected errors [1]. Another difficulty of real-world data sets is that they often contain multiple errors at the same time [19]. We therefore need to consider an ensemble of algorithms to derive suitable data quality rules, while maintaining a short execution time.

The literature review we conducted yielded in the selection of three methods for further investigation. Table 1 places these methods in their methodological categories and shows corresponding literature.

In a first step, we implemented algorithms for each of the four methods and used a small test data set to determine their usefulness for generating data quality rules. An analysis of the results showed promising results for the SVM and Association Rule Learning methods. The Functional Dependencies method suffered from large time and space requirements, which caused the algorithm to

Table 1. Initial selection of methods for detecting data errors.

	Schema-Level	Instance-Level
Dependencies	Functional Dependencies [11,13,18]	Association Rule Learning [3,6]
Multiple Attributes	Support Vector Machine (SVM) [7]	–
Single Attribute	–	–

abort. In a comparative study on established algorithms for discovering functional dependencies Papenbrock et al. found that it is not possible to derive dependencies from a large high-dimensional data set in a limited amount of time [18]. Based on these results we decided to disregard the Functional Dependencies approach and further investigate SVM and Association Rule Learning.

Support Vector Machine (SVM). The general idea of the SVM approach is to utilize the aspect that the material data sets at PharmCo are sparse. Given a tuple of a data set that contains some null and some concrete values we want to label if a value is expected for a certain attribute or not and determine a data quality rule on the schema level. Since a SVM works on numeric attributes, we transformed the given data set to a binary matrix consisting of 0 and 1 values. In this case, a 0 indicates that an attribute contains no value and a 1 indicates that some concrete instance is present. This way, we transferred the data set to a binary classification problem, which are well-suited for SVMs [21]. As SVMs are supervised learning models, a target attribute must be provided. This attribute can either be known as error prone or one of high-importance.

For the SVM analysis we start with a target data set and transform it to a binary matrix. We separate the data to a set of binary samples and a set of binary classes using the target attribute. With these two sets we train a linear SVM and obtain a trained binary classifier. During a migration incoming data is tested against this model. Therefore, we transform an entry of an incoming data set on-the-fly to a binary tuple. This tuple contains all attributes except for the target attribute. As classification result we will retrieve a 0 or 1, which indicates whether this field should be filled or not (see Fig. 3). In case a value is given but the model predicted a 0 or vice versa there is likely an error in the incoming tuple.

The main advantage of this approach is the fast training and classification time. On the contrary, this approach only works for sparse data sets and might therefore not always be useful. Furthermore, the SVM does not explicitly formulate the derived data quality rules, which makes it more difficult to interpret the results.

Association Rule Learning. Our goal with association rule learning is to discover relationships between multiple attributes on an instance level. This means that, in contrast to the SVM approach, we want to identify what values

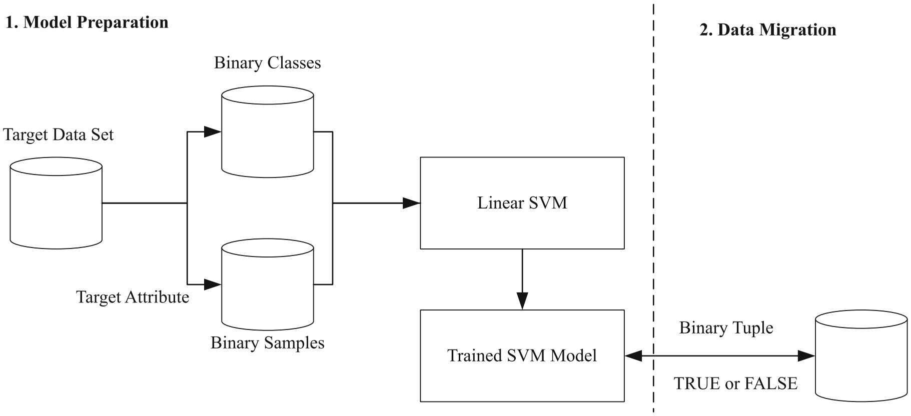

Fig. 3. Functional overview of the SVM method.

often appear together (i.e. frequent itemsets) and build data quality rules using these itemsets. Association rules are well suited for detecting multi-column data errors but are uncommon in professional data quality and migration tools, as these are focused on single column data errors [1,10]. Generally, association rules have the following format:

$$\{AttributeA|ValueA, AttributeB|ValueB \rightarrow AttributeC|ValueC\} \qquad (1)$$

The most well-known algorithm for association rule learning is the Apriori algorithm by Agrawal et al. [3]. Although there are faster solutions available for discovering association rules (e.g. FP-Growth [6]) we decided to use the Apriori algorithm as it is well-established and there are several implementations in different programming languages available.

The Apriori algorithm is an unsupervised ML method that can handle different value types, but not null-values. We therefore filled the missing values in the data set with a fixed ‘NA’ value. The Apriori algorithm furthermore requires a *support* level, which determines how often a value pair needs to appear to be considered frequent and a *confidence* level, which defines how often a rule needs to be true. With these inputs the Apriori algorithm produces a set of association rules. During the data migration we can verify incoming data against the derived set of association rules. If a rule with a high confidence level is not met, we can reason that there is an error in the data set (see Fig. 4).

An advantage of association rule learning is that it is a multivariate method and is not limited to one kind of type. It also produces rules that are easy to understand and interpret for humans. A downside is the complexity of the algorithm and that it suffers from long execution times on data sets with many different or free-text values. Thus, it is vital to pre-process and filter the data from unnecessary attributes to limit the execution times.

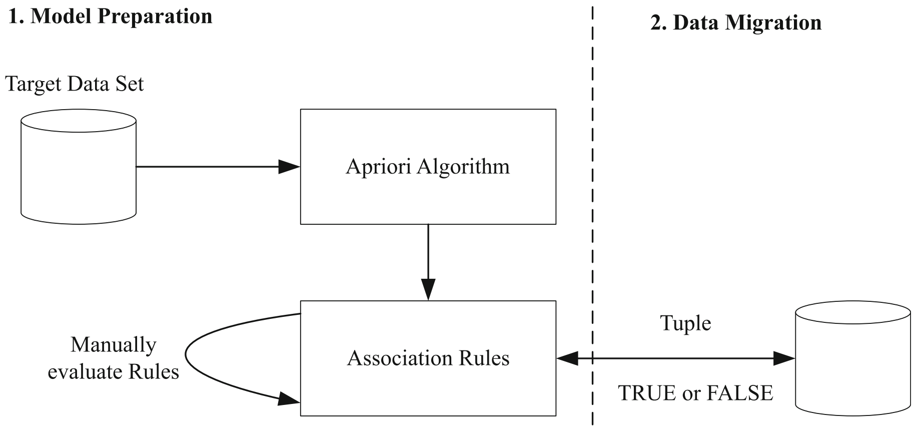

Fig. 4. Functional overview of the Association Rule Learning method.

4 Findings

For evaluating the soundness and suitability of our approach for data migrations we applied the methods to material data sets used in previous data migrations at PharmCo. We decided to use data sets from previous migrations as this offered us the possibility to compare our findings with the ground truth, in which all data quality issues were manually resolved. Specifically, we used a *training set* and a *test set* for the trading goods and spare parts data sets respectively. The *training sets* were old copies of the central SAP system and served as basis for training our ML models. The *test sets* were old copies from a material database of an external subsidiary of PharmCo. This data set simulated the incoming data during a migration. Moreover, we qualitatively evaluated our approach with data migration experts at PharmCo, who helped us to investigate the advantages and disadvantages of the proposed algorithms.

4.1 Support Vector Machine (SVM)

The SVM approach requires a certain target attribute to train a classifier. In discussions with the data migration team at PharmCo we selected three attributes (A: ‘Product hierarchy’, B: ‘Transportation group’, C: ‘Purch Group’) for the trading goods data set and five attributes (A: ‘Product Hierarchy’, B: ‘Gross Weight’, C: ‘Material Type’, D: ‘Lot Size’, E: ‘Valuation class’) for the spare parts data set as target attributes. We decided to train models for these attributes as according to PharmCo they have a tendency for missing values and needed manual review in previous migrations. We trained eight SVM models (S1 to S8) using the *training sets*. For each of these models we used the *test sets* to obtain a classification result and evaluated this against the ground truth to derive the fraction of false positive and false negative classifications. Following the definition of Abedjan et al., we set precision P as the fraction of cells that are correctly marked as errors and recall R as the fraction of actual errors discovered [1]. Table 2 summarizes our results.

Table 2. Evaluation of the SVM approach.

Data Set	*Trading Goods*			*Spare Parts*				
SVM Model	S1	S2	S3	S4	S5	S6	S7	S8
Attribute	A	B	C	A	B	C	D	E
P	0.99	1	0.99	0.99	0.98	0.96	0.99	0.98
R	0.94	0.93	0.96	0.96	0.94	0.9	0.97	0.93

The results show that the SVM approach is highly accurate in correctly detecting potential errors in an incoming data set. Using this approach we can significantly reduce the amount of manual data quality checks that would normally be required during a migration. The SVM approach is particularly useful for sparse attributes that have a tendency for missing values. However, our experiences are based on the two data sets provided by PharmCo. Further evaluations with other data sets should be part of future work to support the generality of the SVM approach for data migrations.

4.2 Association Rule Learning

For association rule learning we conducted several runs of the Apriori algorithm using the *training sets* with different parameter settings (see Table 3). Other than the attributes removed during the pre-processing step (see Sect. 2.4) we did not remove any attribute and conducted the analysis on the remaining 127 and 62 attributes for the trading goods and spare parts data sets respectively. For comparison and evaluation of the different settings we measured the number of rules produced, the execution time and the precision of the rules. We followed the approach of Chiang and Miller for calculating the precision value [8]. They define the precision P of association rule learning to be the fraction of the number of relevant rules over the total number of returned rules. To determine the amount of relevant rules we manually evaluated the derived rules with domain and data migration experts at PharmCo. In this process, we disregarded runs with no rules or with too many rules for manual review. We found the optimal configuration for the trading goods data set at $support >= 90\%$ and $confidence = 100\%$, which produced an output of 43 rules. For the spare parts data set we received an optimum of 14 rules with settings at $support >= 50\%$ and $confidence >= 80\%$.

During the manual review of the derived rules we found that some of the correct and relevant rules have a trivial meaning. For instance, we derived the following rule on dimensionality using the trading goods data set with 100% confidence and 100% support. It specifies that whenever two dimensions have a length of 0.0 the third must also be 0.0.

Table 3. Evaluation of association rule learning.

Data Set	*Trading Goods*				*Spare Parts*			
Settings % (sup. / conf.)	50/70	60/80	80/90	90/100	50/70	50/80	70/90	80/90
# of rules	2399	817	110	43	88	14	0	N/A
Exec. time (Sec.)	102	70	38	16	76	51	48	N/A
P	N/A	N/A	0.85	0.98	0.92	1	N/A	N/A

$$\{Height|0.0, Length|0.0 \rightarrow Volume|0.0\} \tag{2}$$

An example for a more complex rule we derived from the spare parts data set with a support level of 51.4% specifies that whenever a good is measured in kg and its trading is not restricted then its base unit is ‘EA each‘. This rule is true in 96.5% of the cases.

$$\{Weight|\text{KGM kg}, Cat|\text{Y001 Ambient no restrict} \rightarrow BaseUnit|\text{EA each}\} \tag{3}$$

Further to the manual review we evaluated the 43 and 14 rules we derived from the *training sets* against the *test sets* from a previous migration to find potential errors. This test showed that there are no violating tupels in the *test sets*. Nevertheless, this is still a useful result, as we have a validated proof that the data within the *test sets* is correct and do not need an additional manual review. This way we can reduce the amount of manual data quality work.

Overall, the results show that association rule learning is a suitable approach for detecting data errors in data migrations. However, it can be difficult to determine the optimal settings for support and confidence as the results need a manual review. Hereby, a user-friendly explanation of the algorithms and results can help to improve the usability of this approach [4].

4.3 Evaluation

After the case study we conducted a retrospective workshop with data migration experts at PharmCo. The workshop included the author of this paper and four experts from PharmCo. It lasted 90 min and provided valuable insights as the experts could draw on their personal experiences with data migrations. The workshop was structured by the two proposed methods and each one was discussed thoroughly regarding its usefulness and potential downsides.

Overall, we learned that our approach is well received and the prototypical application is still in use. However, a seamless integration into the existing system landscape at PharmCo is necessary for future use. Currently, the data migration team manually implements the rules derived from association rule learning into the existing data migration tool as executable rules. The SVM approach is used for certain attributes that the domain experts consider important. Therefore, the incoming data is tested against the binary classifier using a Python script, that was manually integrated into the data migration tool. PharmCo is planning to extend the current prototype and integrate it with the data migration tool as well as existing databases. This way, a fully integrated and automated tool for data quality rule learning emerges, which helps to simplify cumbersome data migration processes.

5 Related Work

The detection and cleaning of relational data sets (e.g., [1,19]) and the data quality challenges in data migrations (e.g., [16,20]) have both been widely discussed in the scientific literature. Yet, there is only a limited amount of prototypes available that combine both research directions and address data quality issues in data migrations.

For instance, with regard to quality rule learning Shrivastava et al. [22] presented a tool called DQLearn. The tool assists users in the development of data quality rules by providing a method for formalizing a data quality problem in a structured way. This way data quality rules become more explainable and easy to automate. In [4] the authors highlight the need for explainability and customization in automated data quality tools. They argue that the user needs to easily understand and interpret the results provided.

Drumm et al. proposed QuickMig, a system for semi-automatic creation and enforcement of schema mappings [9]. Their approach aims to reduce the complexity of data structures, which helps to lower the data migration efforts. Kaitoua et al. introduced a system called Muses [14]. Muses focuses on supporting data migrations between distributed polystores. Through efficient data reorganizations their system can improve the performance of data migrations by up to 30%. A data dependency graph is used for improving data migrations by Zou et al. [24]. The data dependency graph defines relationships between data components. Using pre-defined compliance criteria an algorithm checks whether a specific instance is migratable or not. This way data consistency and data quality are improved.

Unlike these systems, our approach features several distinct characteristics to support data migrations. (1) We are combining rule detection on a schema and an instance level to identify potential issues on both levels. Other solutions, like QuickMig, are focusing on the schema level [9]. (2) Our approach automatically discovers data quality rules in a limited amount of time. Solutions that utilize data profiling techniques (e.g. [1]) are of limited scalability and therefore not suitable for data migrations. (3) The methods we employ have been tested on real-world data sets. They are flexible to handle common data characteristics, such as sparsity or mixed values.

6 Conclusion

Although data migrations are part of a company's daily business, they are still considered error-prone, expensive and their success rates are low. In our study, we describe the lessons learned from impaired data sets due to data migrations and propose an extended data migration process that ensures data quality. Specifically, we combined a binary SVM classifier and association rule learning to mine data quality rules from a given data set. Incoming data must comply with these rules to be migrated without manual review. These automated checks lead to a reduced amount of manual data quality work and reduced cost. We evaluated both methods against a real-world data set. Our findings showed that both methods produce valuable results and are suitable for an application to data migrations.

With the proposed solution we are addressing the current limitations of data migrations at PharmCo. We created an automated solution that meets the requirements specified by the data migration and domain experts. Most importantly, our tool is capable of deriving data quality rules on the schema and instance level and can therefore prevent different kinds of data errors. Furthermore, the algorithms we used are scalable and have a limited execution time, which makes them suitable for time-critical data migration projects.

Despite the promising results, our study has several limitations. Most importantly, our findings are based on two material data sets. We are therefore planning to evaluate our solution in further data migration scenarios with different data sets and in different companies. This would also support the generality of our findings and help to formalize lessons learned that are generally applicable. It would also be useful to test our approach in a live data migration and investigate the impact our solution has on the performance and the overall migration process. Furthermore, our solution only works with structured data sets. In light of current trends, there is a need to investigate data quality rule generation for migrations of unstructured data sets.

References

1. Abedjan, Z., et al.: Detecting data errors: where are we and what needs to be done? Proc. VLDB Endowment **9**(12), 993–1004 (2016)
2. Abedjan, Z., Golab, L., Naumann, F.: Profiling relational data: a survey. VLDB J. **24**(4), 557–581 (2015)
3. Agrawal, R., Srikant, R., et al.: Fast algorithms for mining association rules. In: Proceedings 20th International Conference Very Large Data Bases, VLDB, vol. 1215, pp. 487–499 (1994)
4. Altendeitering, M., Fraunhofer, I., Guggenberger, T.: Designing data quality tools: findings from an action design research project at Boehringer Ingelheim, pp. 1–16 (2021)
5. Barateiro, J., Galhardas, H.: A survey of data quality tools. Datenbank-Spektrum **14**, 15–21 (2005)
6. Borgelt, C.: An implementation of the FP-growth algorithm. In: Proceedings of the 1st International Workshop on Open Source Data Mining: Frequent Pattern Mining Implementations, pp. 1–5 (2005)
7. Burges, C.J.: A tutorial on support vector machines for pattern recognition. Data Min. Knowl. Disc. **2**(2), 121–167 (1998)

8. Chiang, F., Miller, R.J.: Discovering data quality rules. Proc. VLDB Endowment **1**(1), 1166–1177 (2008)
9. Drumm, C., Schmitt, M., Do, H.H., Rahm, E.: QuickMig: automatic schema matching for data migration projects. In: Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management, CIKM 2007, pp. 107–116. Association for Computing Machinery (2007)
10. Ehrlinger, L., Rusz, E., Wöß, W.: A survey of data quality measurement and monitoring tools. arXiv preprint arXiv:1907.08138 (2019)
11. Fan, W., Geerts, F., Li, J., Xiong, M.: Discovering conditional functional dependencies. IEEE Trans. Knowl. Data Eng. **23**(5), 683–698 (2010)
12. Hipp, J., Güntzer, U., Grimmer, U.: Data quality mining-making a virute of necessity. In: DMKD, p. 6 (2001)
13. Huhtala, Y., Kärkkäinen, J., Porkka, P., Toivonen, H.: TANE: an efficient algorithm for discovering functional and approximate dependencies. Comput. J. **42**(2), 100–111 (1999)
14. Kaitoua, A., Rabl, T., Katsifodimos, A., Markl, V.: Muses: distributed data migration system for polystores. In: 2019 IEEE 35th International Conference on Data Engineering (ICDE), pp. 1602–1605. IEEE (2019)
15. Kruse, S., et al.: Fast approximate discovery of inclusion dependencies. In: Datenbanksysteme für Business, Technologie und Web (BTW 2017) (2017)
16. Matthes, F., Schulz, C., Haller, K.: Testing quality assurance in data migration projects. In: 2011 27th IEEE International Conference on Software Maintenance (ICSM), pp. 438–447 (2011)
17. Morris, J.: Practical data migration. BCS, The Chartered Institute (2012)
18. Papenbrock, T., et al.: Functional dependency discovery: an experimental evaluation of seven algorithms. Proc. VLDB Endowment **8**(10), 1082–1093 (2015)
19. Rahm, E., Do, H.H.: Data cleaning: problems and current approaches. IEEE Data Eng. Bull. **23**(4), 3–13 (2000)
20. Sarmah, S.S.: Data migration. Sci. Technol. **8**(1), 1–10 (2018)
21. Shao, Y.H., Chen, W.J., Deng, N.Y.: Nonparallel hyperplane support vector machine for binary classification problems. Inf. Sci. **263**, 22–35 (2014)
22. Shrivastava, S., Patel, D., Zhou, N., Iyengar, A., Bhamidipaty, A.: DQLearn: a toolkit for structured data quality learning. In: 2020 IEEE International Conference on Big Data (Big Data), pp. 1644–1653. IEEE (2020)
23. Wang, P., He, Y.: Uni-detect: a unified approach to automated error detection in tables. In: Proceedings of the 2019 International Conference on Management of Data, pp. 811–828 (2019)
24. Zou, J., Liu, X., Sun, H., Zeng, J.: Live instance migration with data consistency in composite service evolution. In: 2010 6th World Congress on Services, pp. 653–656. IEEE (2010)