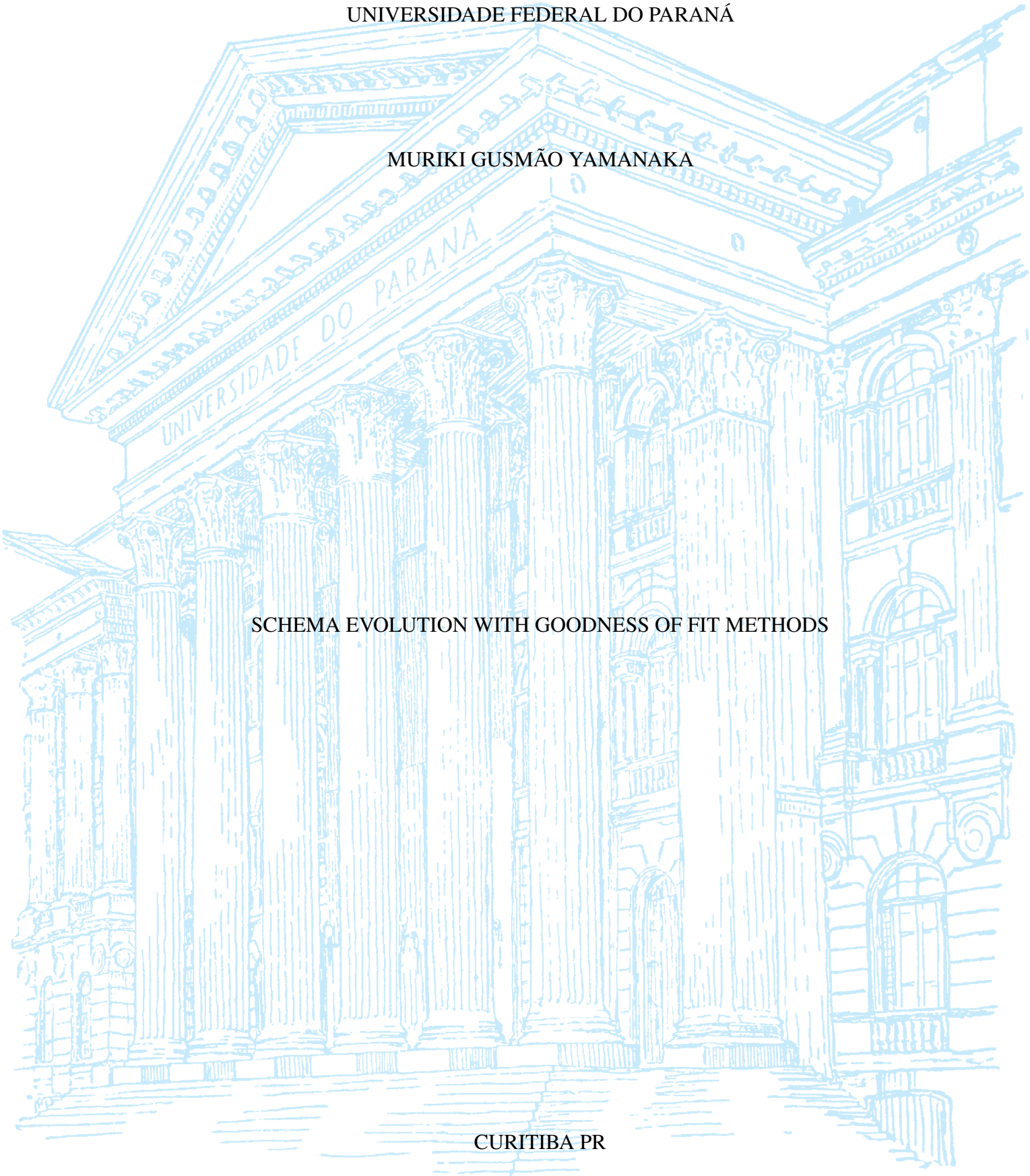
UNIVERSIDADE FEDERAL DO PARANÁ

MURIKI GUSMÃO YAMANAKA



SCHEMA EVOLUTION WITH GOODNESS OF FIT METHODS

CURITIBA PR

2025

MURIKI GUSMÃO YAMANAKA

SCHEMA EVOLUTION WITH GOODNESS OF FIT METHODS

Trabalho apresentado como requisito parcial à conclusão do Curso de Bacharelado em Ciência da Computação, Setor de Ciências Exatas, da Universidade Federal do Paraná.

Área de concentração: *Computação*.

Orientador: Eduardo Cunha de Almeida.

CURITIBA PR

2025

Universidade Federal do Paraná
Setor de Ciências Exatas
Curso de Ciência da Computação

Ata de Apresentação de Trabalho de Conclusão de Curso 2

Título do Trabalho: SCHEMA EVOLUTION WITH GOODNESS OF FIT METHODS

Autor(es):

GRR20203933 Nome: MURIKI GUSMÃO YAMANAKA

GRR_____ Nome:_____

Apresentação: Data: 13/05/2025 Hora: 8h Local: Auditório - Dinf

Orientador: Eduardo Cunha de Almeida

Membro 1: Simone Dominico

Membro 2: Paulo Ricardo Lisboa de Almeida

(nome) (assinatura)

AVALIAÇÃO – Produto escrito		ORIENTADOR	MEMBRO 1	MEMBRO 2	MÉDIA
Conteúdo	(00-40)				
Referência Bibliográfica	(00-10)				
Formato	(00-05)				
AVALIAÇÃO – Apresentação Oral					
Domínio do Assunto	(00-15)				
Desenvolvimento do Assunto	(00-05)				
Técnica de Apresentação	(00-03)				
Uso do Tempo	(00-02)				
AVALIAÇÃO – Desenvolvimento					
Nota do Orientador	(00-20)		**********	**********	
NOTA FINAL		**********	**********	**********	95

Os pesos indicados são sugestões.

Conforme decisão do colegiado do curso de Ciência da Computação, a entrega dos documentos comprobatório de trabalho de Conclusão de Curso 2 deve deve respeitar os seguintes procedimentos: o orientador deve abrir um processo no Sistema Eletrônico de Informações (SEI – UFPR); Selecionar o tipo: *Graduação: Trabalho Conclusão de Curso*; informar os interessados: nome do aluno e o nome do orientador; anexar esta ata escaneada e a versão final do PDF da monografia do aluno; Tramitar o processo para CCOMP (Coordenação de Ciência da Computação).

To my past self, some things went wrong, but many others improved.
To my future self, I hope that everything worked out.

ACKNOWLEDGEMENTS

To my father and sister, who have always supported my decisions, without whom I wouldn't be who I am today.

To C3SL, which helped me develop professionally at university and put everything I learned into practice.

To my teachers Eduardo, Simone and Paulo, who had the patience to correct all my mistakes.

And finally, to all of my friends, with whom I had good lunches and good laughs.

RESUMO

Os dados públicos disponíveis estão sempre sujeitos às novas versões, sendo que cada versão reflete potencialmente alterações aos dados. Essas alterações podem envolver a adição ou remoção de atributos, a alteração de tipos de dados, a modificação de valores ou de sua semântica. A integração desses conjuntos de dados em uma base de dados relacional coloca um desafio significativo: como manter o controle do esquema da base de dados em evolução, enquanto se incorporam diferentes versões das fontes de dados? Este trabalho apresenta uma metodologia estatística para validar a integração de 16 anos de dados abertos do Censo Escolar do Brasil, com uma nova versão lançada anualmente pelo Ministério da Educação Brasileiro (MEC). Vários testes estatísticos da classe *Goodness of Fit* são apresentados em conjunto com uma forma de separação de dados para cada teste de acordo com o número de valores distintos. Outro ponto exibido é como realizar o cálculo da acurácia da comparação entre dois esquemas de dados. Também é mostrado como realizar o tratamento prévio dos dados de entrada para cada teste encontrar atributos de correspondência entre conjuntos de dados de um ano específico e seus potenciais equivalentes em conjuntos de dados de anos anteriores. Os resultados indicam que todos os testes conseguiram corresponder com sucesso colunas de diferentes versões de conjuntos de dados em cerca de 80% dos casos, quando analisadas as 38 colunas de correspondência mais prováveis ao ano anterior.

Palavras-chave: Banco de Dados Relacional. Evolução de Esquema. Integração de Dados. Métodos Estatísticos.

ABSTRACT

Publicly available datasets are subject to new versions, with each version potentially reflecting changes to the data. These changes may involve adding or removing attributes, changing data types, and modifying values or their semantics. Integrating these datasets into a relational database poses a significant challenge: How to keep track of the evolving database schema while incorporating different versions of the data sources? This work presents a statistical methodology to validate the integration of 16 years of open access datasets from Brazil's School Census, with a new version of the datasets released annually by the Brazilian Ministry of Education (MEC). Various statistical tests from the Goodness of Fit class are presented together with a way of separating the data for each test according to the number of distinct values. Another point shown is how the accuracy of the comparison between two data schemas was calculated. It is also shown how the prior processing of the input data is carried out for each test to find matching attributes between datasets from a specific year and their potential equivalents in datasets from previous years. The results indicate that all the tests were able to successfully match columns from different versions of the datasets in around 80% of the cases, when analyzing the 38 most likely matching columns from the previous year.

Keywords: Relational Database. Schema Evolution. Data Integration. Statistical Methods.

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

CDF	Cumulative Distribution Function
IQR	Interquartile Range
CSV	Comma Separated Values
TPCH	Transaction Processing Performance Council - Benchmark
DBMS	Database Management System
LDE	Educational Data Laboratory (*Laboratório de Dados Educacionais*)
MEC	Ministry of Education
IEEE	Institute of Electrical and Electronics Engineers
ISO	International Organization for Standardization
K-S	Kolmogorov–Smirnov
A-D	Anderson–Darling

LIST OF SYMBOLS

χ^2	Chi Square, statistical test

CONTENTS

1 INTRODUCTION

Integrating open data sources is a complex challenge in developing information systems. Open data sources may exhibit structural changes over time when made public, including variations in data types, values, semantics, and missing values, requiring constant evolution of the integrated database schema before the ingestion of new data (Garcia-Molina et al., 2009). The PRISM project, for example, reported an average of 217% schema changes over 48 months across 12 large web information systems (Curino et al., 2009, 2013). The Ensembl Genome project presented over 410 schema versions in 9 years. The Ensembl DB schema contains over 175 individual changes of primary and foreign keys in its schema evolution history.

The evolution of a database schema often leads to mapping errors, compromising the accuracy of stored data and ultimately leading to inconsistencies and inaccuracies in data analysis. Furthermore, differences in data presentation and evolving business needs can significantly hinder the incorporation of new data into existing databases.

In this research, it is introduced a statistical methodology to validate the integration of open-access datasets into the LDE information system. This methodology allows us to track the evolution of the system's database schema across different versions of datasets. The LDE system integrates open-access data from Brazil's School Census to support many studies and public educational policies (Schneider et al., 2023; Alves et al., 2019; Schneider et al., 2020; Silveira et al., 2021). The LDE database contains 17 years of School Census data and is freely accessible. Each year, MEC publishes the School Census[1], which includes comprehensive data from 179,500 schools, such as the number of students, teachers, and classes at each school. However, the publicly available data files have undergone 675 individual changes in naming conventions, as well as the addition and removal of columns over the years. These changes, driven by evolving government requirements, make it challenging for policymakers and researchers to access a unified and integrated reliable source.

Taking the aforementioned challenges in consideration, this work proposes a Goodness-of-fit statistical test approach to evaluate the evolution of the LDE database schema, enhancing the reliability of column matching. Goodness-of-fit tests are meant to define how well some sample of data fits with another given distribution (D'Agostino, 1986). In the context of data integration, the tests conduct data profiling (Abedjan et al., 2015), analyzing column-matching operations such as detecting additions, removals, and changes. This process seeks to minimize errors and inconsistencies in the evolution of an integrated database schema.

Overall, the main contributions with this research are the following:

Quality metrics based on statistical tests for data integration: The methodology encompasses metrics from four Goodness-of-fit statistical tests to evaluate the matches between the continuous and discrete attributes of datasets from different releases to facilitate the integration process. The tests are Kolmogorov-Smirnov test (Berger and Zhou, 2014), Anderson-Darling test (Anderson and Darling, 1952), Chi-Square test (and, 1900) and the G-test (Hoey, 2012).

Analysis of the tests: It is presented the analysis of the results indicating that the methodology can correctly align the columns of different datasets in over 80% of cases considering the Top 38, reducing the amount of manual work required by a data specialist, showing high accuracy and effectiveness in the validation of the integrated schema.

[1]Brazilian School Census Open data (in Portuguese): https://www.gov.br/inep/pt-br/acesso-a-informacao/dados-abertos

Validation of the LDE database schema: The methodology validates the quality of data integrated into the LDE database, thereby supporting the evolution of its schema.

This work is structured as follows: Chapter 2 outlines the changes in the open-access data files, the potential integration problems in a database schema and also discusses related work. Chapter 3 delineates the methodology used in this study. The findings are presented in Chapter 4. In Chapter 5 it is present the results by changing the data separation method. Finally, Chapter 6 provides a summary of the study and outlines the next steps.

2 BACKGROUND AND RELATED WORK

Although schema evolution literature has long acknowledged the complexity of data source integration, the high computational costs associated with general schema evolution techniques have prevented their practical deployment (Cerqueus et al., 2015a; Scherzinger et al., 2016). Schema evolution refers to integrating changes to a data source over time, including adding new sources. Examples of source transformations over time include different column names, changes to the data domain, and their representation. It is also possible for columns or tables to be added as new sources are integrated (Delplanque et al., 2020).

There are many tools to assist in the integration of datasets (a non-exhaustive survey on integration tools is found, here: (Curino et al., 2013)). However, human intervention is often necessary to align open-access datasets containing historical information. This is further complex by the evolution of the open data file structure over time, including changes in column names, value domains, and additions/removals of columns. This research focuses on addressing the challenges associated with column name changes and additions/removals.

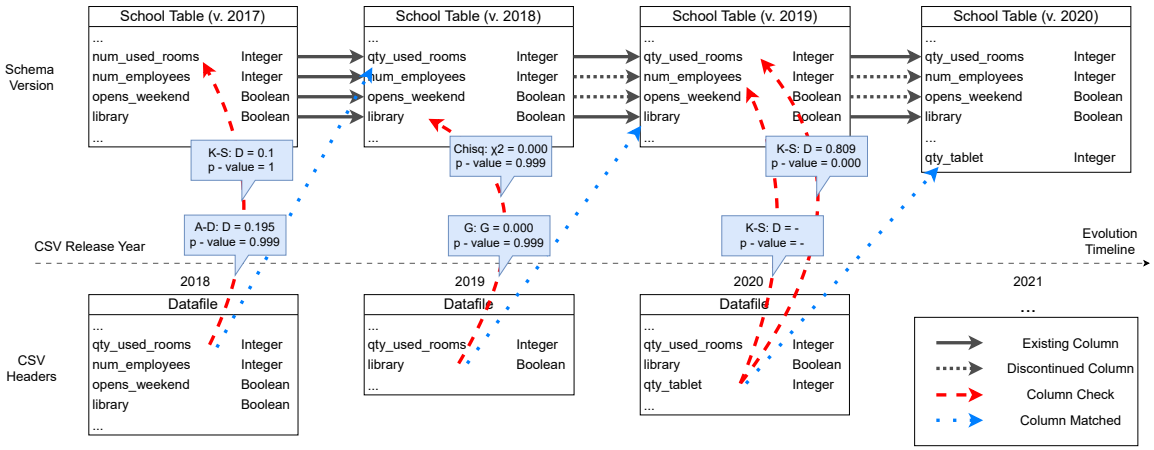


Figure 2.1: Illustration of schema evolution showing the data file headers from 2018 to 2020, as well as the impact of header changes on the integrated schema. Arrows indicating the mappings.

2.1 THE LDE SYSTEM

The LDE system stores data from the School Census over the past 17 years, compiling a vast amount of educational information. Maintaining this dataset is crucial for monitoring trends over time and gaining valuable insights into the Brazilian educational context. Consequently, the LDE system serves as a key resource for leveraging government open data in academic research.

Many projects maintained by MEC and different universities depend on this data, such as the Cost-Student Quality Simulator (SIMCAQ) [1] (Alves et al., 2019). SIMCAQ evaluates the cost of delivering quality education based on various educational and structural variables, such as class size, teacher salaries, and library resources. Another example of a system that depends on the LDE data is MapFor (Schneider et al., 2023), which tracks teachers' academic backgrounds.

[1] https://www.simcaq.c3sl.ufpr.br/ (in Portuguese)

These projects demonstrably impact society, highlighting the importance of maintaining high data quality within the LDE database.

Some of the schema changes over time are illustrated in Figure 2.1. First, to show a change in column names, consider the CSV headers in the year 2018. Originally, in the schema, the column is named "*num_used_rooms*", but when a new data file is released by the MEC open-access files the attribute's name is changed to "*qty_used_rooms*". Secondly, when the 2019 data files are released, dotted arrows that connects schema versions are used to indicate that the attributes *num_employees* and *opens_weekend* are no longer present in that year. Finally, in the 2020 data file, to represent the introduction of a new information, the column "*library*" is show without a arrow connecting it from a previous year.

Properly mapping all these changes in the LDE database is essential to enhancing data quality. In Figure 2.1, consider a scenario where new information is added to the data files of the scholar census, such as "*qty_used_rooms*". Regardless of whether or not this information is already implied in the existing "*num_used_rooms*" column, there might be a tendency to treat it as a new column. This could lead to the addition of a new column to the LDE database ("*qty_used_rooms*"), resulting in schema evolution. However, mapping to this new column can make it difficult to infer existing information ("*num_used_rooms*") without detailed analysis. When a new column is created (such as "*qty_used_rooms*"), instances from previous years are filled with null values, and subsequent analysis may provide incorrect information, failing to indicate that previous data was present in another column (such as "*num_used_rooms*").

2.2 RELATED WORK

Schema evolution management has been the focus of several works over the years. These works conduct empirical investigations into relational schema evolution (Qiu et al., 2013; Vassiliadis et al., 2015). In (Klettke et al., 2017), the authors evaluate schema evolution histories over time, examining data from a data lake and schema versioning. This work analyzes data integration quality and tracks the evolution of the database schema.

Some works have evaluated the schema evolution in the NoSQL database (Meurice and Cleve, 2017; Ringlstetter et al., 2016). In (Cerqueus et al., 2015a), the authors discuss the implementation and customization of verification rules to help developers manage schema evolution and prevent compatibility issues and data loss. (Scherzinger and Sidortschuck, 2020; Scherzinger et al., 2016; Cerqueus et al., 2015b) investigate the evolution of NoSQL database schema, focusing on their flexibility, denormalization practices, and changes over development time through empirical analysis of open-source projects. This work evaluates the quality of schema evolution in a relational database, which implies distinct challenges. The NoSQL schema evolution has greater flexibility and denormalization. However, relational databases enforce constraints and a greater need to maintain integrity.

Prism/Prism++ (Curino et al., 2009, 2013) implements a solution focused on schema evolution in relational databases. Prism uses the data dictionary to track changes in the data schema. It describes an integrated solution to predict and evaluate the impact of schema changes and integrity constraints. The objective is to minimize downtime by automating database migration and documenting schema evolution. Unlike Prism++, which uses a desired schema and the integrity constraints evolution as a input to automate the migration of the data, in this work it is explored a statistical approach to provide as a input only the new data file to be integrated.

The work of Delplanque et al. (Delplanque et al., 2020) discusses the challenges of evolving relational database schema. The authors propose a meta-model approach to automate modifications after database changes, providing recommendations to maintain a consistent state.

As observed in (Etien and Anquetil, 2024), a meta-model for analyzing the impact of changes and ensuring database relational constraints are verified. In contrast, the methodology evaluates the data distribution and other statistical measures without analyzing attribute names. This approach allows us to monitor schema evolution from a data-centric perspective, providing an understanding of how data changes over time.

With the objective of measuring data dependencies in large databases, the work from Piatetsky-Shapiro et al. (Piatetsky-Shapiro and Matheus, 1993) showed pdep (probabilistic dependency), a direct and quantitative measure of how much the knowledge of field X helps to predict the value of field Y. The pdep measure can indicate the direction of the dependence between nominal (discrete and unordered) values and how much the knowledge of one field helps in predicting the other field. In this work, the aim is to make comparisons in numerical data as well as in categorical data.

These related works demonstrate the relevance of the problem and show the need for methods to provide support for data migration in relational models to mitigate problems in maintaining the integrity of a system and minimizing downtime.

3 GOODNESS-OF-FIT SCHEMA EVOLUTION METHODOLOGY

In this chapter, it is present the statistical methodology used in the integration of open-access datasets into the LDE database. The methodology employs Goodness-of-fit statistical tests to match the columns of the CSV files released each year with the existing columns in the database. First, in Section 3.1, the tests are defined in the context of schema evolution tests. In Section 3.2, it is presented how to determine which test will be performed over data. Finally, in Section 3.3, both the matching and the accuracy algorithms are defined, which uses specific metrics given by the Goodness-of-fit tests to determine the correct match of each column for a given year.

3.1 GOODNESS-OF-FIT

The main hypothesis is that Goodness-of-fit statistical tests ensure reliable *data quality* metrics during column matching. These tests provide information about *data distributions*, *means*, *variances*, and *magnitude* of observed differences. Among the tests it is used the well-known Kolmogorov–Smirnov test, Anderson–Darling test, Chi-Square test, and the G (Log likelihood ratio) test to compare the distributions of a column in a given year with possible matches from the next year.

Let $x : (x_1, x_2, \ldots, x_m)$ and $y : (y_1, y_2, \ldots, y_n)$ be the distributions (collected data between years) being compared of sizes m and n, respectively. Now, each test is briefly described.

Kolmogorov–Smirnov test: This test verifies if two samples are statistically similar. In this methodology, it determines whether the data in two columns from different years follow the same distribution. This allows the evaluation of data consistency over time by comparing the base year with the following year based on the distribution of the samples. Let F_m and G_n be the empirical Cumulative Distribution Functions (CDFs) for the x and y samples defined as follows:

$$F_m(t) = \frac{\text{number of sample x' } \leq t}{m} \tag{3.1}$$

$$G_n(t) = \frac{\text{number of sample y' } \leq t}{n} \tag{3.2}$$

the Kolmogorov-Smirnov test is defined as follows:

$$D = max|F_m(t) - G_n(t)|, min(x, y) \leq t \leq max(x, y) \tag{3.3}$$

where samples are considered to come from the same distribution if D is small enough (D'Agostino, 1986; Berger and Zhou, 2014).

Considering the example illustrated by Figure 2.1, the attributes "num_used_rooms" and "qty_used_rooms" present the same distribution and data type. In this particular case, the K-S test shows the $D = 0.1$ and $p - value = 1$.

Anderson–Darling test: Similar to the Kolmogorov-Smirnov test, the Anderson–Darling test considers the differences between the distributions, with the difference that this test gives more weight to the tails of the distributions when compared to the Kolmogorov-Smirnov. For comparing two distributions, the Anderson–Darling statistic can be computed as follows:

$$A^2 = \frac{1}{N(mn)} \sum_{j=1}^{N-1} \frac{(NX_j - jm)^2 + (NY_j - jn)^2}{j(N-j)} \tag{3.4}$$

where $N = m + n$, $Z_1 < \cdots < Z_N$ is the pooled ordered sample, and X_j and Y_j are the number of observations in x and y that are not greater than Z_j, respectively (Pettitt, 1976).

The Anderson–Darling test applied to "num_used_rooms" and "qty_used_rooms" results in a statistic of $A^2 = 0.195$ and $p - value = 0.999$, also indicating a statistically similarity in their distributions. This suggests that both the columns are likely to be compatible.

Chi-Square Test (χ^2): The Chi-Square test is a classical test for goodness of fit problems, some of the advantages of using this test is that it is well adapted for the case when the distribution function of a sample is discontinuous i.e., represents a discrete distribution, and it is known how to adapt the statistic for the case when parameters of the distribution must themselves be estimated from the sample (and, 1974).

The Goodness-of-fit Chi-Square test is typically applied to nominal variables (such as labels without inherent ordering). In this test, the observed counts of observations in each category are compared with the expected counts, which are calculated based on a theoretical expectation (McDonald, 2014). Here, this test is applied to evaluate categorical data, including Boolean attributes.

The formula is:

$$\chi^2 = \sum_i \frac{(O_i - E_i)^2}{E_i} \tag{3.5}$$

Where χ^2 is the Chi-Square statistic, O_i represents the observed frequency for each category and E_i represents the expected frequency for each category.

In Figure 2.1 the test is illustrated by comparing the Boolean attribute "$library$" in 2018 schema version against the attribute of same name in 2019 datafile. The test result shows that this comparison have a $\chi^2 = 0.000$ and $p - value = 0.998$, hence it is inferable that the expected counts in schema version in the year 2018 have a very similar distribution from the observed counts from 2019 datafile. In the same way, the larger the difference between observed and expected, the larger the test statistic becomes.

G-test (Log likelihood ratio): Very similar to the Chi-Square test, the G-test is used to determine whether the number of observations in each category fits a theoretical expectation, particularly when the sample size is large and the variables are nominal. Although both tests yield similar results, the Chi-Square test is the most commonly used in such scenarios. G-tests, on the other hand, are a subclass of likelihood ratio tests, which are a general category of tests with various applications for assessing the fit of data to mathematical models. Consequently, the G-test can facilitate more elaborate statistical analyses (McDonald, 2014).

Furthermore, the more O_i and E_i are different, the less well this approximation will work, and Chi-Square will tend to compute erroneous answers. The effects of a single outlier in a small sample set will be more pronounced, which explains why the Chi-Square often fails in situations with little data (Hoey, 2012).

The formula is:

$$G = 2 \sum_i O_i \cdot \ln\left(\frac{O_i}{E_i}\right) \tag{3.6}$$

Where G is the Log Likelihood Ratio statistic, O_i represents the observed frequency and E_i represents the expected frequency. In Figure 2.1, applying the test to the same columns "$library$" it is obtained the statistical result $G = 0.000$ and a $p - value = 0.998$, which, as mentioned above, has very similar results to Chi-Square test.

3.2 TEST SELECTION DELIMITER

Kolmogorov-Smirnov and Anderson Darling tests are designed for continuous data distributions while Chi-Square and G-tests are for discrete data distributions. This fact lead the experiments to the following problem: When wanting to compare a column from the LDE database and a column from the new arriving data file, how to determine which test will be more appropriate? Applying the K-S test, for example, in a Boolean column such as "$library$" could lead to wrong results since the test takes into account the maximum distance between the Cumulative Distribution Functions. Similarly, when applying the Chi-Square to a continuous data such as "num_used_rooms" the test will have to create too many categories for each value and could also result in the wrong tests being performed.

Here the hypothesis is that by using the count of distinct values the data can give an approximation to which test should be used i.e., if the distinct count of both datas when comparing the database to the data file are below a certain value, then the test to be applied is Chi-Square and G-test. On the other hand, if both counts are above that same value, then the applied test will be K-S and A-D. In order to generate this value, care had to be taken to choose it based on another data set, since using the same data to classify oneself could generate a bias in the results. To evaluate this hypothesis it is used the well known TPCH database (Transaction Processing Performance Council, 1999) to generate the delimiter value. The TPCH database schema consists of eight tables that represent a simplified model of a business environment, typically focused on a retail or wholesale business. The schema is designed to support complex queries and decision-making processes.

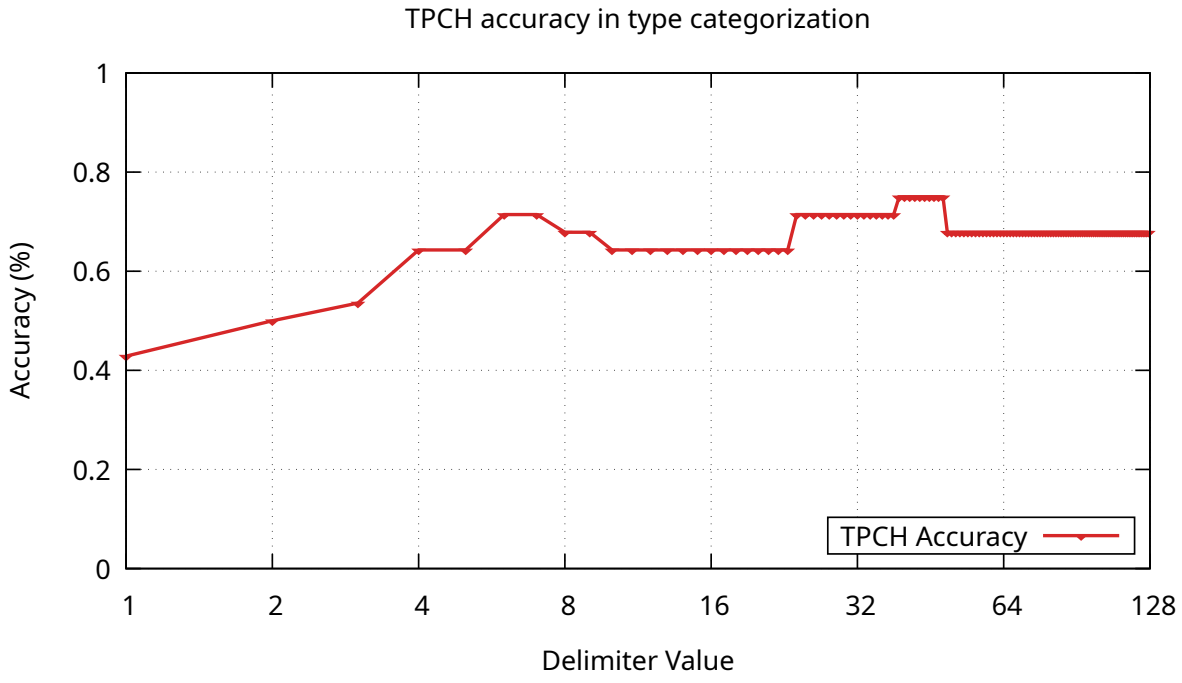


Figure 3.1: The accuracy to determine if a column is Numerical or Categorical using TPCH database (Log scale)

In order to find the best delimiter value in TPCH, Figure 3.1 exhibits the accuracy in data labeling. Specifically, it shows how many columns were correctly classified as numerical or categorical based on whether the x-axis (delimiter) ranges from 1 to 128. Based on the results, the value 40 can be used initially as a thumb rule to make the distinction of which test will be performed against which columns. This is shown in algorithm 1, whenever a comparison is made between the database (s_col) and the new data file (d_col) it is verified if both data belongs to the same group (line 6-9). The algorithm returns a set with statistical values and the $p - value$ from each comparison, respectively.

Algorithm 1: DATACOMPARE($database$, $schema\_cols$, $datafile$, $datafile\_cols$, $delimiter$)

Input:
$database$: Database data;
$schema\_cols$: Names of the columns in database;
$datafile$: New data to be integrated in database;
$datafile\_cols$: Names of the columns in datafile;
$delimiter$: Value to determine which test will be performed over which columns.
Result: A map with the results of tests comparison.

```
1  result_set = empty_set
2  for s_col ∈ schema_cols do
3      for d_col ∈ datafile_cols do
4          data1 = distinct(database[s_cols])
5          data2 = distinct(datafile[d_cols])
           // Check which test will be performed
6          if (length(data1) ≤ delimiter) ∧ (length(data2) ≤ delimiter) then
7              result = add(chisq(database[s_col], datafile[d_col]))
8              result = add(Gtest(database[s_col], datafile[d_col]))
9          else if (length(data1) > delimiter) ∧ (length(data2) > delimiter) then
10             result = add(KS(database[s_col], datafile[d_col]))
11             result = add(AD(database[s_col], datafile[d_col]))
```

Result: result_set

As mentioned earlier, already knowing which value would be the best possible delimiter for the given database could bias the results. Therefore, the delimiter from all the results generated in Section 4 are only derived from the TPCH database. Furthermore, in Chapter 5, a comparison is presented between the results in Section 4 and those considering delimiter 8 (the best delimiter for the LDE database) or the type defined in the database management system.

3.3 THE SCHEMA MATCHING ALGORITHM

The Algorithm 2 describes how the new data file is integrated into the database and how the accuracy is calculated, returning the accuracy result of each test at the end. Initially, the $result\_set$ of the comparisons made by Algorithm 1 are filtered by the threshold value, so that if the $p - value$ from a comparison exceeds a specified threshold (line 2), the algorithm will only operate on the columns from the later year as a potential match and discard all other comparisons.

The comparison of column matches falls under the broader domain of data profiling, which involves analyzing columns (Abedjan et al., 2015; Pena et al., 2021). In data profiling, the number of potential column comparisons can grow exponentially with the number of attributes in a relation. While our algorithm inherits this complexity, it focuses on the specific task of

comparing two columns, resulting in a worst-case scenario of quadratic complexity when dealing with identical schemas.

Most importantly, the algorithm enables the classification of data columns into three categories to guide integration decisions: continued columns (lines 5-9), missing columns (lines 10-12), and new columns (lines 13-15). For the integration forecast, only the $result\_set$ and $p\_value\_threshold$ parameters are necessary. The parameters $continued\_cols$, $missing\_cols$ and new_cols are used for accuracy calculation. Finally, the top parameter is used to generate a ranking of potential matches.

Identical columns exhibit consistent data across different years, hence a comparison between these two columns is expected to generate a high $p - value$. Given that the threshold is already filtered, the continuing columns are considered the columns from the database that still have comparisons remaining in the $result\_set$. If a column from the database has no match after the filter, then it is considered as a missing data column. In the same way, if a column from the data file has no matches, then it is considered as a new column. For each column tested i.e., the set of continued, missing and new columns, whenever an evolution is correctly predicted a counter *hit_count* is incremented, at the end of each test the counter is divided by the number of columns to generate the final result.

Algorithm 2: MATCHACCURACY($result\_set$, $continued\_cols, missing\_cols, new\_cols, p\_value\_threshold, top$)

Input:
$result\_set$: Algorithm 1 result;
$continued\_cols$: List of columns present in both database and data file;
$missing\_cols$: List of columns present only in database;
new_cols: List of columns present only in data file;
$p\_value\_threshold$: Threshold to determine $p - value$ limit;
top: Top ranking.
Result: Accuracy from each test.

1 $accuracy\_result$ = empty_set
// Remove all matches with p-value lesser than the threshold
2 $result\_set = result\_set[p\_value \geq p\_value\_threshold]$
3 **for** $test \in \{KS, AD, ChiSq, G\}$ **do**
4 $\quad hit\_count = 0$
$\quad$ // Calculate continued columns accuracy
5 $\quad$ **for** $col \in continued\_cols$ **do**
6 $\quad\quad match\_set = sort(result\_set[statistic])$
7 $\quad\quad matches = head(match\_set[col], top)$
8 $\quad\quad$ **if** $col \in matches$ **then**
9 $\quad\quad\quad hit\_count = hit\_count + 1$
$\quad$ // Calculate missing columns accuracy
10 $\quad$ **for** $col \in missing\_cols$ **do**
11 $\quad\quad$ **if** $length(result\_set[col]) = 0$ **then**
12 $\quad\quad\quad hit\_count = hit\_count + 1$
$\quad$ // Calculate new columns accuracy
13 $\quad$ **for** $col \in new\_cols$ **do**
14 $\quad\quad$ **if** $length(result\_set[col]) = 0$ **then**
15 $\quad\quad\quad hit\_count = hit\_count + 1$
16 $\quad hit\_count = hit\_count/(length(continued\_cols) + length(missing\_cols) + length(new\_cols))$
17 $\quad accuracy\_result$ = add($test, hit\_count$)

Result: accuracy_result

4 EXPERIMENTAL RESULTS

In this chapter, the experiments conducted using the previously presented statistical techniques on data retrieved from the LDE database are delved into. The experimental protocol followed is described, and the results obtained from the evaluation process are reported. It is provided the access to data and the complete source code of the LDE system. [1]

4.1 EXPERIMENTAL PROTOCOL

It is used the R implementation of the Goodness-of-fit methods described in Section 3.1. Instead of feeding the columns data directly to the Algorithm 1, which could lead to problems when estimating the $p - value$ (indicating the confidence of a given column to be a correct match), first it is performed a data pre-processing to clean some dirty data that could naturally come from the census.

For the K-S and A-D tests first all the NULL data is removed. Secondly, it is necessary to perform an outlier removal. For this it is employed the Interquartile Range (IQR) method for outlier detection by finding the first (Q1) and third (Q3) quartiles, representing 25% and 75% of the original data, respectively. The IQR is the difference between Q1 and Q3. The outliers are identified and removed as values falling below Q1 subtracted by 1.5 times the IQR, or above Q3 added by 1.5 times the IQR. Afterwards, the remaining data is organized into a *10-bin* histogram and normalized. The input data for the R functions used in the tests will consist of this *10-bin* histogram.

The data pre-processing for the Chi-Square and G-tests are different. Instead of a NULL removal, the missing data is labeled as a new category. Whenever two data vectors are compared, the maximum value between both vectors is identified and incremented by one. Then, all NULL values in the vectors are replaced with this incremented maximum value. As explained in Section 3.1, both tests generate a frequency table of all the categories found in data. Due to how R implements the Chi-Square and G-test functions, if a comparison is made between different frequency tables (i.e. different sizes or different categories) the function will return an error. To overcome this problem, both frequency tables are filtered by the intersection between both table categories. Afterwards the frequency table is normalized so that the distribution of values is between 0 and 1. This last step is highly important, especially when considering data accumulated over time, as will be described in section 4.3. At the end of this process the data is used as the input for the R functions.

Finally, in Algorithm 2, for the $p\_value\_threshold$ parameter the value is defined as $p\_value\_threshold = 0.9$ (i.e., $\alpha = 0.1$), which is a common practice when accepting/rejecting the NULL hypothesis (i.e., the columns come from the same distribution).

4.2 RESULTS – MATCHES CONSIDERING THE PREVIOUS YEAR

Given that the MEC School Census is released every year, whenever a new data file must be integrated in the database, the match comparisons is made by the previous year. In Table 4.1,

[1] LDE system: https://dadoseducacionais.c3sl.ufpr.br/ (in Portuguese).

the results are presented considering the accuracy of the Top 1, where in Algorithm 2, it is represented by the top parameter.

Here, a successful match is defined as an exact match between a column and its corresponding column from the previous year. In Table 4.1, column *Year* defines the reference year, which is necessary to match the columns with the previous year. The column *Changes* shows $[x]c$ as the number of changed, $[y]+$ as the number of new, and $[z]-$ as the number of removed columns when compared with the previous year (considering the ground-truth). The numerical and categorical columns represent the number of attributes of each type in the database up to the respective year. Lastly, the K-S, A-D, Chi-Sq and G-test columns represents the results returned by Algorithm 2.

For example, considering 2022 in Table 4.1. This year, when analyzing the official data made available from MEC and comparing it with the previous year (2021), 2 columns changed their names; 88 new columns appeared, presenting data that was not collected in the previous year, and 17 columns disappeared, presenting data that was no longer collected.

Year	Changes	Numeric Columns	K–S Test	A-D Test	Changes	Categorical Columns	Chi-Sq Test	G-Test
2007	-	-	-	-	-	-	-	-
2008	*no change*	7	0.900	0.500	[3]c [4]+ [0]−	107	0.317	0.317
2009	*no change*	7	0.700	0.600	[1]c [9]+ [2]−	114	0.434	0.425
2010	*no change*	7	0.600	0.400	*no change*	114	0.450	0.459
2011	*no change*	7	0.455	0.545	[0]c [3]+ [2]−	115	0.469	0.469
2012	*no change*	7	0.727	0.818	[1]c [22]+ [0]−	137	0.459	0.459
2013	[0]c [11]+ [0]−	18	0.632	0.526	[0]c [15]+ [14]−	138	0.424	0.424
2014	*no change*	18	0.550	0.400	[1]c [0]+ [0]−	138	0.555	0.547
2015	[17]c [1]+ [0]−	19	0.520	0.440	[122]c [32]+ [7]−	163	0.265	0.259
2016	*no change*	19	0.435	0.348	*no change*	163	0.465	0.465
2017	*no change*	19	0.391	0.435	*no change*	163	0.459	0.459
2018	[3]c [0]+ [15]−	4	0.783	0.739	[0]c [0]+ [1]−	162	0.453	0.453
2019	[0]c [18]+ [2]−	20	0.571	0.619	[9]c [4]+ [22]−	144	0.383	0.383
2020	[0]c [9]+ [0]−	29	0.500	0.533	[0]c [71]+ [0]−	215	0.129	0.120
2021	[0]c [45]+ [0]−	74	0.493	0.547	[2]c [0]+ [27]−	189	0.386	0.391
2022	[0]c [31]+ [0]−	105	0.308	0.206	[2]c [57]+ [17]−	229	0.294	0.294
2023	[0]c [52]+ [0]−	157	0.403	0.279	[0]c [17]+ [4]−	242	0.271	0.259
Average (stdev)			0.560 (0.158)	0.496 (0.158)	Average (stdev)		0.388 (0.107)	0.386 (0.109)

Table 4.1: Accuracy considering the Top 1 and the results year by year.

As one can observe in Table 4.1, although the Kolmogorov–Smirnov (K-S) test presented the best results when considering the averaged results, it is still far from the ideal. For few columns such as the year 2008 the K-S test resulted in a good accuracy of 90%, but in year 2011 the value decreased to 45%. Also, when a more complex scenario appeared (i.e. a scenario where a database administrator would have difficulty integrating data without a data dictionary), such as the year 2022 with 263 columns present in database (74 numerical and 189 categorical columns) and 334 columns in the data file (105 numerical and 229 categorical columns), the precision decreased even further (reaching about 30% only).

However, during the tests, an interesting phenomenon is observed: even when the column is not perfectly fitted with the previous year in Top 1, the correct fit still presented a high probability (according to each test) of belonging to its proper fit. This is due to the high similarity of the distributions between different columns, such as "$drinking\_water$" and "$filtered\_water$" in the LDE database. Although they do not represent the same information, it is noteworthy that a school with drinking water is likely to also have filtered water.

In light of this, it is also conducted tests by increasing the Top value (i.e. considering a hit if the predicted fit appears among the N most probable fits for a given approach) to a point where even if the Top value were further increased, the results wouldn't change significantly (more

than 1%). Increasing the Top value can present a more realistic scenario than the Top 1 since it can show the most probable fits to a specialist, who will choose the correct one according to the best of their domain knowledge. In Bellahsene research (Bellahsene et al., 2011) it is explained that not always there is a correct set of matches or mappings between a source and a target schema. The expected answer depends not only on the semantics, but also on the transformation that the mapping designer was intending to make. Hence, many evaluations of matching or mapping tools are performed by human experts.

The Table 4.2 shows that all the tests had an average accuracy over 80% when considering the results for the Top 38 value. As can be seen, unlike the Top 1 results, even though numerous changes have taken place over the years, the tests have still managed to maintain high accuracy. For years 2012 and 2018, both K-S and A-D achieved 100% accuracy when integrating the data.

In addition, although the Top 1 results for Chi-square and G-test were worse than K-S and A-D, for the Top 38 the scenario is reversed. Even though the number of columns worked on is much higher (more than 100 columns by 2019 and more than 200 by 2020) the average accuracy of both tests was 85%. Furthermore, for the years 2010, 2016 and 2017, when there was no evolution in the data schema, both tests were able to predict data integration with 100% accuracy.

Year	Changes	Numeric Columns	K–S Test	A-D Test	Changes	Categorical Columns	Chi-Sq Test	G-Test
2007	-	-	-	-	-	-	-	-
2008	*no change*	7	1.000	0.900	[3]*c* [4]+ [0]−	107	0.923	0.923
2009	*no change*	7	0.900	0.900	[1]*c* [9]+ [2]−	114	0.885	0.885
2010	*no change*	7	0.800	0.900	*no change*	114	1.000	1.000
2011	*no change*	7	0.909	0.909	[0]*c* [3]+ [2]−	115	0.938	0.947
2012	*no change*	7	1.000	1.000	[1]*c* [22]+ [0]−	137	0.774	0.774
2013	[0]*c* [11]+ [0]−	18	0.789	0.632	[0]*c* [15]+ [14]−	138	0.768	0.768
2014	*no change*	18	0.900	0.900	[1]*c* [0]+ [0]−	138	0.993	0.993
2015	[17]*c* [1]+ [0]−	19	0.760	0.760	[122]*c* [32]+ [7]−	163	0.578	0.584
2016	*no change*	19	0.913	0.870	*no change*	163	1.000	1.000
2017	*no change*	19	0.913	0.957	*no change*	163	1.000	1.000
2018	[3]*c* [0]+ [15]−	4	1.000	1.000	[0]*c* [0]+ [1]−	162	0.962	0.962
2019	[0]*c* [18]+ [2]−	20	0.857	0.810	[9]*c* [4]+ [22]−	144	0.784	0.784
2020	[0]*c* [9]+ [0]−	29	0.700	0.667	[0]*c* [71]+ [0]−	215	0.627	0.627
2021	[0]*c* [45]+ [0]−	74	0.613	0.693	[2]*c* [0]+ [27]−	189	0.865	0.870
2022	[0]*c* [31]+ [0]−	105	0.570	0.542	[2]*c* [57]+ [17]−	229	0.694	0.694
2023	[0]*c* [52]+ [0]−	157	0.701	0.675	[0]*c* [17]+ [4]−	242	0.888	0.888
	Average (stdev)		0.833 (0.135)	0.820 (0.141)	Average (stdev)		0.855 (0.137)	0.856 (0.136)

Table 4.2: Accuracy Considering the Top 38 and the results year by year.

This result shows that these Goodness-of-fit tests used in combination with the proposed Algorithms (Algorithms 1 and 2) can significantly decrease the manual work of the domain's specialist, who will find the correct match in the first proposals of the algorithm (38 possible matches) instead of needing to find the proper fit considering all possible columns available for a given year (the number of columns in a datafile).

The Figure 4.1 illustrates how the increase of the Top number impacted the results of each test. The accuracy percentage represents the average of the results obtained over the years for each Top value. As presented earlier, Top 1 shows a very low accuracy, while Top 38 shows a considerably better average. As can be seen, even though it's not the best possible result when increasing the Top value, considering the 10 most probable fits, both tests have already shown practically the same accuracy in comparison with Top 38. Even the K-S and A-D tests have stabilized in the Top 5.

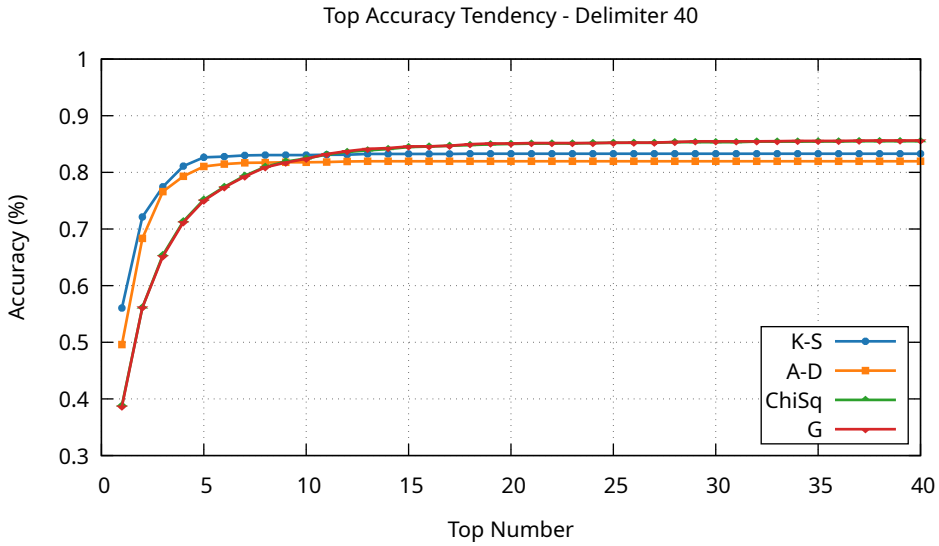


Figure 4.1: Top accuracy tendency

4.3 RESULTS – MATCHES CONSIDERING THE ACCUMULATED YEARS

In this Section, it is followed the same protocol as in Section 4.2, with the difference that when trying to match the nth reference year, all the data from the first year to the $nth - 1$ year are used to create the distribution to be compared. For instance, when trying to match the reference year of 2010, the data distribution from 2010 is compared with the distributions of the years 2007, 2008, and 2009 combined.

To make this possible, whenever a test is performed there is a normalization step in the data pre-processing. For the K-S and A-D tests, each bin is divided by the number of tuples used to create the histogram. In contrast, for the Chi-Square and G-tests, each category in the frequency table is divided by the total number of tuples. The concept of this test is that by grouping more data to be compared, the closer it gets to the real underlying distribution of the data.

However, over the years there was a decrease in accuracy, resulting in an average of 25% accuracy for both K-S and A-D tests. On the other hand, the Chi-Square and G-test methods, although still highly accurate, had their average results reduced to 76%. It is hypothesized that external factors, such as data collection policy changes over time, made it more difficult to make a correct match when including the data from previous years in the distribution for comparison.

5 DIFFERENT VALUES FOR DELIMITER

Having the delimiter calculated by TPCH might lead to a wrong categorization in the data as numeric or categoric, resulting in the wrong test being performed over columns. Taking this into account, if the optimal value were previously determined and the right tests were performed perfectly on the right columns, one could say that the results in Top accuracy might improve. To validate this hypothesis, it is also carried out the previous tests with different delimiter values as follows.

5.1 BEST VALUE FOR LDE DATABASE

Considering that it is viable to previously determine if a column is categorical or numerical (e.g. having a minimal data dictionary) it would be possible to determine the optimal value for the LDE database. Analyzing which value would suit the most, the Figure 5.1 shows the accuracy achieved for each delimiter value varying from 1 to 128. As one can observe, the delimiter 8 is the value where more than 90% of the database columns are correctly labeled (i.e., numerical as numerical and categorical as categorical).

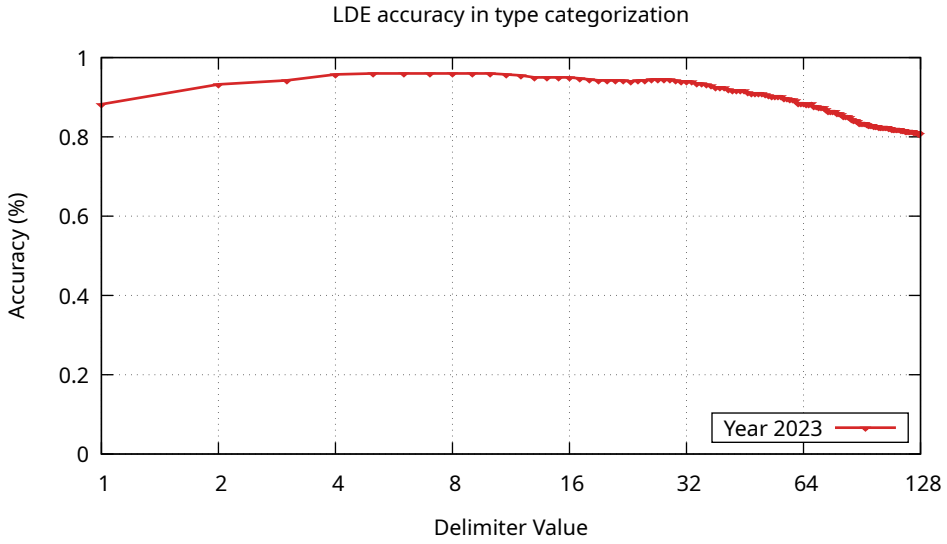


Figure 5.1: Optimal value to separate data between numerical or categorical, calculated from the LDE database (Log scale)

As Figure 5.2 shows, even if some of the columns are wrongly labeled i.e., the Chi-Square and G-test perform over numerical data or the K-S and A-D tests perform over categorical data, the tests are consistent enough (higher than 80%) to give the same proximation from the previous results to which data should be matched.

5.2 SMALLEST INTEGER TYPE IN A DBMS

One pattern that can be observed in the LDE database is that most of the categorical columns have either Boolean or Tinyint (1 Byte) as their data type in the DBMS. Based on this observation,

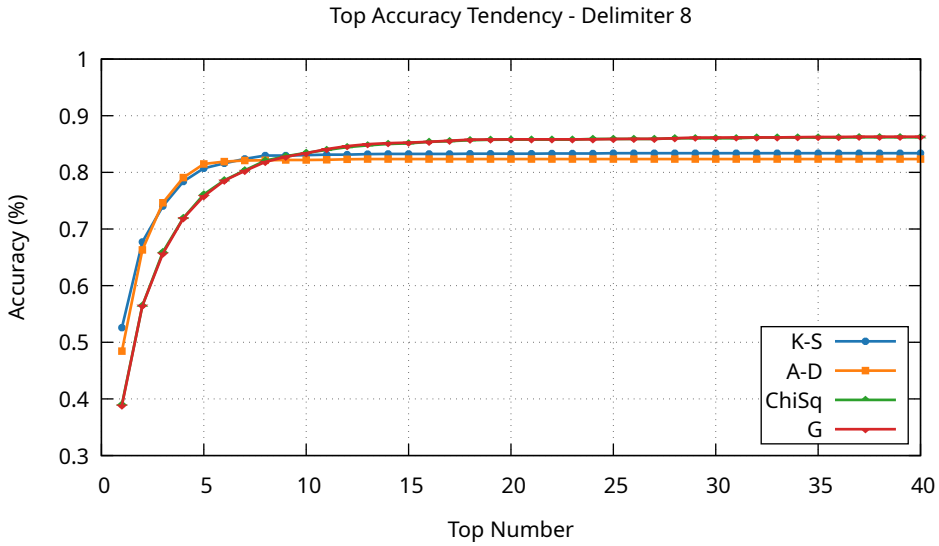


Figure 5.2: Top accuracy tendency using delimiter 8 (Best fit for LDE)

it is also conducted tests with the delimiter value of 127, which is the maximum value the Tinyint type can represent. In this situation, as can be observed in Figure 5.1, many of the columns that were originally numerical will be wrongly labelled as categorical, and the Chi-Square test, along with the G-tests, will be performed over them. Once more, the Figure 5.3 shows that the Top stabilization still reaches around 80% of accuracy, even though this time the value used for the separation is far different from the optimal value and the frequency tables are much bigger.

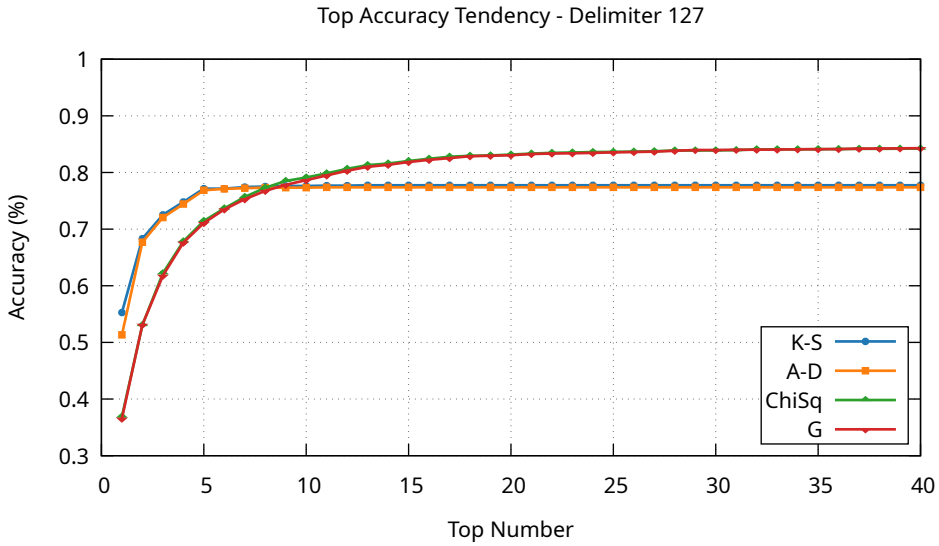


Figure 5.3: Top accuracy tendency using delimiter 127 (Tinyint maximum representation)

6 CONCLUSION AND FUTURE WORK

This work presents a methodology for identifying matches between attributes of the census datasets from a given year and their possible matches in a dataset released in a subsequent year. It is hypothesized that using statistical tests to evaluate the evolution of the LDE database schema enhances the reliability of column matching. Indeed, the methodology finds the matching attributes and show what the changes are, such as adding new data or data columns not present in the year evaluated. Results showed that the approach significantly reduced the manual effort required by a specialist.

Although the results for Top 1 were very poor, when the average accuracy was analyzed from Top 10 onwards, all the tests showed values above 80%. Even in situations where the new data had many changes such as different names, new columns or a lack of columns and the number of columns in the database was considerably large (more than 100 columns), the results still managed to stay around 70%. There have even been years in which the tests were 100% correct for the Top 38.

To solve the problem of determining which test will be performed on which columns, it is used a delimiter of the number of distinct values, generated from the TPCH database. Although the delimiter does not perfectly separate the data for the appropriate statistical tests, it is shown that the variation of this delimiter, both for the ideal value and for the largest representable value in Tinyint, does not significantly impact the results, reducing them by less than 5% for the K-S and A-D tests, and by 2% for the Chi-Square and G-tests.

Another point to mention is how drastically the results were affected when considering the cumulative years. Although it is common to think that using more data might be favorable to data comparison, the results showed that the accuracy was drastically reduced.

6.1 FUTURE WORK

In further work, it is intended to include matching other types of data frequently found in databases, such as real numbers (e.g., IEEE 754 Floating point values), text (e.g., ISO text and varchar), dates and hours. Decimal numbers can be worked on with the K-S and A-D tests by considering a different previous treatment of the data. For text, measures such as LCS (Longest Common Subsequence) can be used to quantify the difference between two texts. Finally, quantifying the difference between dates and hours can be a more complex problem due to the frequency with which the data is integrated.

In addition, it is also necessary to develop a way of separating the types of data to be worked on, i.e. how to differentiate a column between the different new types of categories that a value can represent, which test is the most appropriate and how much the results are impacted by the separation error.

Another possible point to explore is altering the match algorithm to take into account not only the results of a single test for the evolution of the schema, but to use something similar to a weighting system to take into account the results of multiple tests in an attempt to increase the accuracy of the results.

6.2 PUBLICATIONS

This work was presented and published at the 2024 Brazilian Database Symposium (Simpósio Brasileiro de Banco de Dados - SBBD) under the title “Statistical Validation of Column Matching in the Database Schema Evolution of the Brazilian Public School Census” (Yamanaka et al., 2024), presenting 4 statistical tests for the integration of numerical data. Since then, the tests have been reworked and now include tests for both numerical and categorical data, as well as a way of separating the tests presented together with the TPCH database.

REFERENCES

Abedjan, Z., Golab, L., and Naumann, F. (2015). Profiling relational data: a survey. *VLDB J.*, 24(4):557–581.

Alves, T., Silveira, A. A. D., Schneider, G., and Fabro, M. D. D. (2019). Financiamento da escola pública de educação básica: a proposta do simulador de custo-aluno qualidade. *Educação E Sociedade (in Portuguese)*, 40.

and, K. P. (1900). X. on the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 50(302):157–175.

and, M. A. S. (1974). Edf statistics for goodness of fit and some comparisons. *Journal of the American Statistical Association*, 69(347):730–737.

Anderson, T. W. and Darling, D. A. (1952). Asymptotic Theory of Certain "Goodness of Fit" Criteria Based on Stochastic Processes. *The Annals of Mathematical Statistics*, 23(2):193 – 212.

Bellahsene, Z., Bonifati, A., Duchateau, F., and Velegrakis, Y. (2011). *On Evaluating Schema Matching and Mapping*, pages 253–291. Springer Berlin Heidelberg, Berlin, Heidelberg.

Berger, V. W. and Zhou, Y. (2014). Kolmogorov–smirnov test: Overview. *Wiley statsref: Statistics reference online*.

Cerqueus, T., de Almeida, E. C., and Scherzinger, S. (2015a). Safely managing data variety in big data software development. In *1st IEEE/ACM BIGDSE*, pages 4–10.

Cerqueus, T., Scherzinger, S., and de Almeida, E. C. (2015b). Controvol: Let yesterday's data catch up with today's application code. In *WWW Companion*, pages 15–16.

Curino, C., Moon, H. J., Deutsch, A., and Zaniolo, C. (2013). Automating the database schema evolution process. *VLDB J.*, 22(1):73–98.

Curino, C., Moon, H. J., and Zaniolo, C. (2009). Automating database schema evolution in information system upgrades. In *2nd ACM HotSWUp 2009*.

D'Agostino, R. (1986). *Goodness-of-Fit-Techniques*. Statistics: A Series of Textbooks and Monographs. Taylor & Francis.

Delplanque, J., Etien, A., Anquetil, N., and Ducasse, S. (2020). Recommendations for evolving relational databases. In *CAiSE 2020*, pages 498–514.

Etien, A. and Anquetil, N. (2024). Automatic recommendations for evolving relational databases schema. *arXiv preprint arXiv:2404.08525*.

Garcia-Molina, H., Ullman, J. D., and Widom, J. (2009). *Database systems - the complete book (2. ed.)*. Pearson Education.

Hoey, J. (2012). The two-way likelihood ratio (g) test and comparison to two-way chi squared test.

Klettke, M., Awolin, H., Störl, U., Müller, D., and Scherzinger, S. (2017). Uncovering the evolution history of data lakes. In *IEEE BigData 2017*, pages 2462–2471.

McDonald, J. H. (2014). *Handbook of Biological Statistics*. Sparky House Publishing, Baltimore, Maryland, 3rd edition.

Meurice, L. and Cleve, A. (2017). Supporting schema evolution in schema-less nosql data stores. In *24th IEEE SANER*, pages 457–461.

Pena, E. H. M., de Almeida, E. C., and Naumann, F. (2021). Fast detection of denial constraint violations. *Proc. VLDB Endow.*, 15(4):859–871.

Pettitt, A. N. (1976). A two-sample anderson-darling rank statistic. *Biometrika*, 63(1):161–168.

Piatetsky-Shapiro, G. and Matheus, C. J. (1993). Measuring data dependencies in large databases. In *Proceedings of the 2nd International Conference on Knowledge Discovery in Databases*, AAAIWS'93, page 162–173. AAAI Press.

Qiu, D., Li, B., and Su, Z. (2013). An empirical analysis of the co-evolution of schema and code in database applications. In *ACM SIGSOFT*, page 125–135.

Ringlstetter, A., Scherzinger, S., and Bissyandé, T. F. (2016). Data model evolution using object-nosql mappers: folklore or state-of-the-art? In *2nd IEEE/ACM BIGDSE*, page 33–36.

Scherzinger, S., de Almeida, E. C., Cerqueus, T., de Almeida, L. B., and Holanda, P. (2016). Finding and fixing type mismatches in the evolution of object-nosql mappings. In *EDBT/ICDT Workshops*, volume 1558.

Scherzinger, S. and Sidortschuck, S. (2020). An empirical study on the design and evolution of nosql database schemas. In *Conceptual Modeling*, pages 441–455. Springer Inter. Publishing.

Schneider, G., Gallotti Frantz, M., and Alves, T. (2023). Infraestrutura das escolas públicas no brasil: desigualdades e desafios para o financiamento da educação básica. *Revista Educação Básica em Foco (in Portuguese)*, 17(2).

Schneider, G., Silveira, A. A., and Alves, T. (2020). Mapeamento da formação de docentes no paraná: um olhar para o indicador de adequação. *Jornal de Políticas Educacionais (in Portuguese)*, 1(3).

Silveira, A. D., Schneider, G., and Alves, T. (2021). *Simulador de Custo-Aluno Qualidade (SimCAQ): Trajetória e Potencialidades.* Inep/MEC (in Portuguese).

Transaction Processing Performance Council (1999). Tpc-h: A benchmark for decision support. Accessed: 2025-04-29.

Vassiliadis, P., Zarras, A. V., and Skoulis, I. (2015). How is life for a table in an evolving relational schema? birth, death and everything in between. In *Conceptual Modeling*, pages 453–466.

Yamanaka, M., de Almeida, D., de Almeida, P. R., Dominico, S., Peres, L., Sunye, M., and Almeida, E. (2024). Statistical validation of column matching in the database schema evolution of the brazilian public school census. In *Anais do XXXIX Simpósio Brasileiro de Bancos de Dados*, pages 498–509, Porto Alegre, RS, Brasil. SBC.