

Measuring discord among multidimensional data sources

Alberto Abelló

Universitat Politècnica de Catalunya
Barcelona, Spain
aabello@essi.upc.edu

James Cheney

University of Edinburgh
Edinburgh, Scotland
jcheney@inf.ed.ac.uk

ABSTRACT

Data integration is a classical problem in databases, typically decomposed into schema matching, entity matching and record merging. To solve the latter, it is mostly assumed that ground truth can be determined, either as master data or from user feedback. However, in many cases, this is not the case because firstly the merging processes cannot be accurate enough, and also the data gathering processes in the different sources are simply imperfect and cannot provide high quality data. Instead of enforcing consistency, we propose to evaluate how *concordant* or *discordant* sources are as a measure of trustworthiness (the more discordant are the sources, the less we can trust their data).

Thus, we define the *discord measurement* problem in which given a set of uncertain raw observations or aggregate results (such as case/hospitalization/death data relevant to COVID-19) and information on the alignment of different data (for example, cases and deaths), we wish to assess whether the different sources are concordant, or if not, measure how discordant they are.

1 INTRODUCTION

Scientists often analyse data by placing different indicators (e.g., number of patients or number of deaths) in a multidimensional space (e.g., geography and time). The multidimensional model and OLAP tools [2] have been used in the last 30 years with this purpose, as a more powerful and structured alternative to spreadsheets. However, despite these being mature technologies, problems like managing missing and contradictory information are still not solved.

OLAP tools are typically used in data warehousing environments where consistent and well known data go through a well structured cleaning and integration process merging different sources. Nevertheless, in the wild, sources are typically incomplete and not well aligned, and such data cleaning and integration processes are far from trivial, resulting in imperfect comparisons. For example, different actors often report measures at different granularities that can only be compared after aggregation or cleaning. On doing this, even if the aggregation performed is correct, due to reporting mistakes, mereological discrepancies, or incompleteness, it could happen that the indicator of the whole (e.g., cases at country level) is different from the aggregation of indicators of its parts (e.g., states, districts), if these come from a different source (or even from the same source). Like in the parable of the blind men describing an elephant after touching different parts of its body (i.e., touching the trunk, it is like a thick snake; the leg, like a tree stump; the ear, like a sheath of leather; the tail tip, like a furry mouse; etc.), in many areas like epidemiology, different data sources reflect the same reality in slightly different and partial ways. This challenge is well-illustrated by

COVID-19 data where missing, incomplete, or inconsistent numbers have been blamed for bad decision making and unnecessary suffering [22].

Thus, in such a complex context, it is necessary to have a tool that precisely measures discrepancies for the available data. Indeed, [6] and [16] measure the differences in the descriptive multidimensional data and their structure. Instead, we aim at evaluating the reliability of the numerical indicators, given some required alignment declaration (e.g., aggregation or scale correction). At this point, it is important to highlight that, even if some work like [24] proposes to treat textual data as indicators (allowing to aggregate them too), we restrict ourselves to numerical measures, whose discrepancies cannot be evaluated using string similarity metrics like the ones surveyed in [27]. These would rather be part of a preliminary step of entity matching over dimensional descriptors.

Contributions. Incomplete information is typically handled in relational databases by using NULL values. However, it is well known that NULLs are overloaded with different meanings such as nonexisting, unknown or no-information [3]. Thus, we propose to use NULL only for nonexisting or no-information, and enrich the data model with *symbolic variables* that allow to represent the partial knowledge we might have about unknown numerical values. While using symbolic variables for NULLs is not a new idea, introduced for example in classical models for incomplete information such as c-tables and v-tables [18] and more recently in data cleaning systems such as LLUNATIC [15], our approach generalizes unknowns to be arbitrary (linear) expressions that in the end define a setting for the evaluation of the trustworthiness of different sources of multidimensional data based on their concordancy/discordancy using standard linear or quadratic programming solvers. More concretely, in this paper we contribute by defining the problem of *discord measurement* of databases under some merging processes.

Organization. Section 2 presents a motivational example that helps to identify the problem defined in Section 3, whose solution is then exemplified in Section 4. The paper concludes with the related work and conclusions in Sections 5 and 6.

2 RUNNING EXAMPLE

A real application of our approach to COVID-19 is available in an extended version of this paper [1], but for illustration purposes, we provide here a fictitious running example of discordance evaluation. Let's consider a scenario where a network of actors (i.e., governmental institutions) take *primary* measurements of COVID-19 cases and *derive* some aggregates from those. We illustrate how to model this scenario using database schemas and views, and describe the different problems we need to solve.

Example 2.1. The statistical institute of *Panem* (a fictional country) generates census reports on the weekly excess of deaths (assumed attributable to COVID-19) in the country. Since we are just considering *Panem*, we can model this information using a

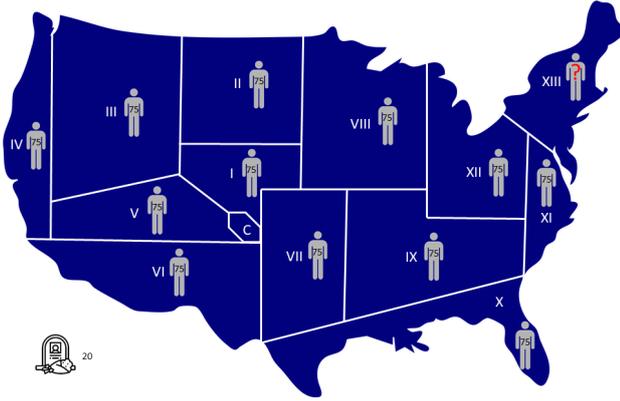


Figure 1: Panem's map with epidemiological information

table (primary key underlined> indicating the number declared per week:

```
Census(week, deaths)
```

We would have greater trust in multiple consistent reports of the same quantity if they had been obtained independently; if they all came from a single source, we would be more skeptical. Therefore, in an epidemiological verification process, we gather different complementary sources of information providing surrogates or approximations to the desired measurements.

Example 2.2. Suppose that *Panem*, as depicted in Fig. 1, comprises thirteen districts (D_I, \dots, D_{XIII}). In each district, there are several hospitals and a person living in D_i is monitored by at most one hospital. Hospitals report their number of cases of COVID-19 to their district governments, and each district government reports to the Ministry of Health (MoH).

Given their management autonomy, the different districts in *Panem* use different and imperfect monitoring mechanisms and report separately the COVID-19 cases they detect every week. Despite being gathered at health facilities, *Panem* is only reporting to the Centre for Disease Prevention and Control (CDC) partial information at the district level and the overall information of the country. We can model this using relational tables with the weekly district and country information.

```
ReportedDistrict(district, week, cases)
ReportedCountry(week, cases)
```

In an idealized setting, we would expect to know all the relationships and have consistent measurements for each primary attribute, and each derived result would be computed exactly with no error. However, some relationships may be unknown and both primary and derived attributes can be noisy, biased, unknown or otherwise imperfect.

Example 2.3. The following view aggregates the district-level for each week, which should coincide with the values per country:

```
CREATE VIEW AggReported AS
SELECT week, SUM(cases) AS cases
FROM ReportedDistrict GROUP BY week;
```

Moreover, it is already known that COVID-19 mortality depends on the age distribution and vaccination status of the population, but let us assume an average Case-Fatality Ratio (CFR) of 1.5% which is reasonable for an unvaccinated population. In terms of SQL, we would have the following view which estimates

the number of deaths based on the number of reported cases in the country.

```
CREATE VIEW Inferred AS
SELECT week, 0.015 * cases AS deaths
FROM ReportedCountry;
```

Example 2.4. Ideally, if all COVID-19 cases were detected, and we knew the exact CFR as well as the effects of the pandemic in other causes of death, the week should unambiguously determine the number of cases and deaths (i.e., information derived from reported cases, both at district and country levels, and mortality in the census must coincide). In terms of SQL, these constraints could be checked using assertions like the following.

```
CREATE ASSERTION SumOfCases CHECK (NOT EXISTS
(SELECT * FROM ReportedCountry r JOIN AggReported a
ON r.week=a.week WHERE r.cases <> a.cases));
CREATE ASSERTION NumberOfDeaths CHECK (NOT EXISTS
(SELECT * FROM Census c JOIN Inferred i
ON c.week=i.week WHERE c.deaths <> i.deaths));
```

Thus, we see that SQL already provides the required mechanisms to freely align the different sources and impose the coincidence of values. Nevertheless, as explained above, achieving exact consistency seems unlikely in any real setting. Indeed, using existing techniques it is possible to check consistency among data sources when there is no uncertainty, but it is not straightforward, in the presence of unknown NULL values or suspected error in reported values, to determine whether the various data sources are consistent with the expected relationships.

Example 2.5. It is easy to see that the following database is not consistent with our view specification, in part because the cases of a district (i.e., D_{XIII}) are not reported, but also the second assertion is violated (i.e., too many people died—20—compared to the inferred number based on the cases reported and the CFR—only 15).

```
ReportedDistrict("I", "2110W25", 75)
...
ReportedDistrict("XII", "2110W25", 75)
AggReported("2110W25", 900)
ReportedCountry("2110W25", 1000)
Inferred("2110W25", 15)
Census("2110W25", 20)
```

Indeed, using existing mechanisms, we can easily detect the problem (i.e., assertion violations). However, we cannot measure how far are the data from really being consistent. For example, the country reporting a thousand cases would violate as many assertions as if reporting one million, but its degree of inconsistency with the other sources is absolutely different.

3 PROBLEM FORMULATION

We aim at extending DBMS functionalities for accurate concordancy evaluation in the presence of overlapping sources for the same numerical data. Given on the one hand the queries and views specifying the expected behavior (a.k.a. alignment of sources), and on the other the data corresponding to observations of some of the inputs, intermediate results, or (expected) outputs, is the observed numeric data complete and concordant considering the alignment specification? If there is missing data, can the existing datasets be extended to some complete instance that is concordant? Finally, how far from being fully consistent are the numerical data?

Given such an *idealized* scenario (specified by its schema and views) and a collection of actual observations (both primary and derived), we can still consider two different problems:

- (A) *Value estimation*: estimate the values of numerical attributes of interest (e.g., the number of cases and deaths across *Panem*) that make the system consistent.
- (B) *Discord evaluation*: Evaluate how far is the actual, discordant dataset from an idealized concordant one.

Problem (A) is the well-studied statistical estimation problem. However, many sources behave as black boxes, and it can be very difficult to precisely quantify the uncertainty and underlying assumptions in many situations, especially where the interrelationships among different data sources are complex. Instead, we consider problem (B). Given a (probably incomplete but overlapping) set of instances, we assume only a merging process specification in the form of *expectations* about their alignment, expressed using database queries and views. Our goal in this paper is not to find a realistic estimate of the true values of unknown or uncertain data, but instead to quantify how close the data are to our expectations under the given alignment. It is important to clarify that while the approach we will adopt does produce estimates for the uncertain values as a side-effect, they are not guaranteed to have any statistical validity unless additional work is done to characterize the sources of uncertainty, which we see as a separate problem.

Therefore, the key contribution of this paper is that both checking concordance and measuring discord can be done by augmenting the data model with *symbolic expressions*, and this in turn can be done consistently and efficiently in a RDBMS with the right set of algebraic operations. Indeed, we define and measure the degree of discordance of different data sources with complementary multidimensional information, where uncertainty may arise from NULLs standing for unknown values, or reported measurements that have some unknown error. To do so, we need to (1) define a variant of relational algebra for queries over (sets of) finite maps represented as *symbolic tables*, (2) formally define the concordance and discordance problems, and (3) show that they can be solved by reduction to linear or quadratic programming, respectively.

4 PROPOSED SOLUTION BY EXAMPLE

The basic idea in this paper is to represent *unknown* real values with variables, which can occur multiple times in a table, or in different tables, representing the same unknown value, and more generally unknown values can be represented by symbolic (linear) expressions in $\mathbb{R}[X]$. However, key values used in key-fields are required to be known. This reflects the assumption that the source database is *partially closed* [14], that is, we assume the existence of master data for the keys (i.e., all potential keys are coincident and known).

Definition 4.1. A *symbolic table*, or *s-table* $R : K \triangleright V$ is a table (with the name prepended with \textcircled{S}) in which key attributes K are mapped to discrete non-null values and value attributes V are mapped to symbolic expressions in $\mathbb{R}[X]$.

Suppose we are given an ordinary database instance, which may have missing values (i.e., NULLs) and uncertain values (i.e., reported values which we do not believe to be exactly correct). To allow such situation, we replace values with symbolic expressions containing variables. This can be done in many ways, with

different justifications based on the application domain. For example, we can replace uncertain values v with " $v \cdot (1 + x)$ " (or simply x if $v = 0$) where x is an error variable. On the other hand, to handle NULLs in s-tables we simply replaced each NULL with a distinct variable.

Example 4.2. It is easy to see that there are many possibilities of assigning cases of COVID-19 to the different districts of *Panem* that add up to 1,000 per week, and consequently improve the consistency of our database, which may be easily represented by replacing constants by symbolic expressions " $75(1 + x_i)$ ", where x_i is an error parameter representing that cases may be missed or overreported in every district. The cases for district D_{XIII} , that were not reported at all, could then be simply represented by a variable x_{XIII} . On the other hand, we also know that attributing all the excess deaths to COVID-19 involves some imprecision, so we should apply some error term " $(1 + z)$ " to the numbers coming from the census, too. Nevertheless, this may not completely explain the mismatch between cases reported at the country level and deaths, and there might also be some doubly-counted or hidden cases in *Panem* (for example in the Capitol which is assumed not to have any cases), which we represent by variable " $(1 + y)$ ". Therefore, s-tables $\textcircled{S}ReportedDistrict : \{district, week\} \triangleright \{cases\}$, $\textcircled{S}ReportedCountry : \{week\} \triangleright \{cases\}$ and $\textcircled{S}Census : \{week\} \triangleright \{deaths\}$ would contain:

```

 $\textcircled{S}ReportedDistrict (" I " , " 2110W25 " , 75 * (1 + x_I) )$ 
...
 $\textcircled{S}ReportedDistrict (" XII " , " 2110W25 " , 75 * (1 + x_{XII}) )$ 
 $\textcircled{S}ReportedDistrict (" XIII " , " 2110W25 " , x_{XIII} )$ 
 $\textcircled{S}ReportedCountry (" 2110W25 " , 1000 * (1 + y) )$ 
 $\textcircled{S}Census (" 2110W25 " , 20 * (1 + z) )$ 

```

Example 4.3. Given the s-tables in Example 4.2, the SQL views in Section 2 can be algebraically expressed as:

```

 $\textcircled{S}AggReported := \gamma_{\{week\};\{cases\}}(\textcircled{S}ReportedDistrict)$ 
 $\textcircled{S}Inferred := \epsilon_{deaths:=0.015*cases}(\textcircled{S}ReportedCountry)$ 

```

Where $\gamma_{K;V}$ is an aggregation operation that sums V grouping by K ; and $\epsilon_{V_{new}:=f(V_{old})}$ derives a new value attribute as a (linear) function of the pre-existing ones.

We represent the expected relationships between source and derived data using a generalization of view specification called *alignment specifications*. Alignment specifications may define derived s-tables as the fusion of multiple views, written $\textcircled{S}R \sqcup \textcircled{S}T$. The fusion operator combines the information from two s-tables, resulting in an s-table with constraints that ensure that the values reported for common keys in $\textcircled{S}R$ and $\textcircled{S}T$ are equal.

Example 4.4. Given the s-tables in Example 4.2 and queries in Example 4.3, the SQL assertions in Example 2.4 can be specified as:

```

 $\textcircled{S}SumOfCases := \textcircled{S}ReportedCountry \sqcup \textcircled{S}AggReported$ 
 $\textcircled{S}NumberOfDeaths := \textcircled{S}Census \sqcup \textcircled{S}Inferred$ 

```

The discord is, intuitively, the shortest distance between the actual observed, uncertain data and a hypothetical concordant database instance that is consistent given the constraints introduced by the alignment specification. The more distant from any such concordant instance, the more discordant our data are. Then, the degree of discordance of our database given an alignment and according to a distance metric (a.k.a., cost function) equals the solution to the quadratic programming problem formed by

minimizing the metric subject to the constraints introduced by coincident instances in different sources found on fusing them with \sqcup operation.

Example 4.5. From the specification in Example 4.4, we get the constraints represented by the following system of equations:

$$\begin{aligned} 1000(1+y) &= 75(1+x_1) + \dots + 75(1+x_{12}) + x_{13} \\ 0.015 * 1000(1+y) &= 20 * (1+z) \end{aligned}$$

Obviously, even considering only positive values for the different variables, that system has many solutions. One solution S_1 consists of taking all x_i to be zero, $y = -0.1$ and $z = -0.325$. This corresponds to assuming there is no error in the twelve districts' reports and there are no cases in District XIII. Another solution S_2 sets $x_I = \dots = x_{XII} = 0$ and $x_{XIII} = 100$, then $y = 0$ and $z = -0.25$ which corresponds to assuming D_{XIII} has all of the missing cases. Of course, whether S_1 or S_2 (or some other solution) is more plausible depends strongly on domain-specific knowledge. Nevertheless, given a *cost function* assigning a cost to each solution, we can compare different solutions in terms of how much correction is needed (or discord exists). For example, we might consider a cost function that simply takes the sum of the squares of the variables:

$$c_1(\vec{x}, y, z) = \left(\sum_{i \in \{I, \dots, XIII\}} x_i^2 \right) + y^2 + z^2$$

Using this cost function, S_1 has cost ≈ 0.116 while S_2 has cost 10000.0625, the first solution is much closer to being concordant, because a large change to x_{XIII} is not needed. Alternatively, we might give the unknown number of cases in D_{XIII} no weight, reflecting that we have no knowledge about what it might be, corresponding to the cost function

$$c_2(\vec{x}, y, z) = \left(\sum_{i \in \{I, \dots, XII\}} x_i^2 \right) + y^2 + z^2$$

that assigns the same cost to S_1 but assigns cost 0.0625 to S_2 , indicating that if we are free to assign all unaccounted cases to x_{XIII} then the second solution is closer to concordance.

Besides alternatives in the cost function, we could weight variables considering the reliability of the different districts as well as the central government, and the historical information of the census. However, these values depend on knowledge of the domain and we will leave exploration of more sophisticated cost functions to future work.

5 RELATED WORK

The problems described above are related to Consistent Query Answering (CQA) [12], which tries to identify the subset of a database that fulfills some integrity constraints, and corresponds to the problem of identifying certain answers under open world assumption [5]. In CQA, distance between two database instances is captured by symmetric difference of tuples. However, in our case, the effects of an alignment are not only reflected in the presence/absence of a tuple, but also in the values it contains. This leads to the much closer Database Fix Problem (DFP) [7, 9], which aims at determining the existence of a fix at a bounded distance measuring variations in the numerical values.

Both DFP as well as CQA become undecidable in the presence of aggregation constraints. Nonetheless, these have been used to drive deduplication [11]. However, our case is different since we are not questioning correspondences between entities to create aggregation groups, but instead trying to quantify their (in)consistency in the presence of complex transformations.

Another known result in the area of DFP is that prioritizing the repairs by considering preferences or priorities (like the data sources in our case) just increases complexity. An already explored idea is the use of where-provenance in the justification of the new value [15], but with pure direct value imputation (without any data transformation). In contrast, we consider that there is not any master data, but multiple contradictory sources, and we allow aggregates, while [15] only uses pure equalities (neither aggregation nor any real arithmetic) between master and target DBs.

From another perspective, our work is related to incompleteness in multidimensional databases, which has been typically focused on the problems generated by imprecision in hierarchical information [13], [6] and [16]. Only more recently, attention has shifted to missing values in the measures. Bimonte et al. [8] presents a linear programming-based framework that imputes missing values under some constraints generated by sibling data at the same aggregation level, as well as parent data in higher levels. We could consider this a special case of our approach, where there is a single data source and alignment is predefined.

The setting we have described shares many motivations in common with previous work on provenance. The semiring provenance model [17] is particularly related, explaining why why-provenance [10] is not enough (e.g., in the case of alternative sources for the same data) and we need how-provenance to really understand how different inputs contribute to the result. They propose the use of polynomials to capture such kind of provenance. Further, Amsterdamer et al. [4] extended the semiring provenance model to aggregations by mixing together annotations and values, but the fine-grained provenance information may become prohibitively large. However, to the best of our knowledge no practical implementations exist. As noted earlier, our s-tables are similar in some respects to c-tables studied in incomplete information databases [18]. Our data model and queries is more restricted in some ways, due to the restriction to finite maps, and the fact that we do not allow for conditions affecting the presence of entire rows, but our approach supports aggregation, which is critical for our application area and which was not handled in the original work on c-tables.

There have been implementations of semiring provenance or c-tables in systems such as Orchestra [19], ProQL [20], ProvSQL [25], and Mimir [23], respectively. In Orchestra provenance annotations were used for update propagation in a distributed data integration setting. ProQL and ProvSQL implement the semiring model but do not allow for symbolic expressions in data or support aggregation. Mimir is a system for querying uncertain and probabilistic data based on c-tables; however, in Mimir symbolic expressions and conditions are not actually materialized as results, instead the system fills in their values with guesses in order to make queries executable on standard RDBMSs. Thus, Mimir's approach to c-tables would not suffice for our needs since we need to generate the symbolic constraints for the QP solver to solve. On the other hand, our work shows how some of the symbolic computation involved in c-tables can be implemented in-database.

We have reduced the concordancy evaluation problem to quadratic programming, a well-studied optimization problem. Solvers such as OSQP [26] can handle systems of equations with thousands of equations and variables. However, we have not made full use of the power of linear/quadratic programming. For example, we could impose additional linear inequalities on unknowns to constrain that certain error or null values have to be

positive or within some range. Likewise, we have defined the cost function in one specific way but quadratic programming permits many other cost functions to be defined, for example with different weights for each variable or with additional linear cost factors.

As noted in Section 2, we have focused on the problem of evaluating concord/discord among data sources and not on using the different data sources to estimate the actual values being measured. It would be interesting to extend our framework by augmenting symbolic tables and queries with a probabilistic interpretation, so that the optimal solution found by quadratic programming produces statistically meaningful consensus values (similarly to the work of Mayfield et al. [21]).

6 CONCLUSIONS

In many real settings, such as epidemiological surveillance, ground truth is not known or knowable and we still need to integrate discordant data sources with different levels of trustworthiness, completeness and self-consistency. In this setting without any master data, we still would like to be able to measure how close the observed data is to our idealized expectations. Thus, we proposed definitions of *concordance* and *discordance* capturing respectively when data sources we wish to fuse are compatible with one another, and measuring how far away the observed data are from being concordant. Consequently, we can compare discordance measurements over time to understand whether the different sources are becoming more or less consistent with one another.

Our approach to symbolic evaluation of multidimensional queries appears to have further applications which we plan to explore next, such supporting other forms of uncertainty expressible as linear constraints, and adapting our approach to produce statistically meaningful estimates of the consensus values.

ACKNOWLEDGMENTS

The work of Alberto Abelló has been done under project PID2020-117191RB-I00 funded by MCIN/AEI/10.13039/501100011033. The work of James Cheney was supported by ERC Consolidator Grant Skye (grant number 682315).

REFERENCES

- [1] Alberto Abelló and James Cheney. 2022. Eris: Measuring discord among multidimensional data sources (extended version). arXiv:2201.13302 [cs.DB]
- [2] Alberto Abelló and Oscar Romero. 2018. Online Analytical Processing. In *Encyclopedia of Database Systems, Second Edition*, Ling Liu and M. Tamer Özsu (Eds.). Springer. https://doi.org/10.1007/978-1-4614-8265-9_252
- [3] Serge Abiteboul, Richard Hull, and Victor Vianu. 1995. *Foundations of Databases*. Addison-Wesley. <http://webdam.inria.fr/Alice>
- [4] Yael Amsterdamer, Daniel Deutch, and Val Tannen. 2011. Provenance for aggregate queries. In *ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS)*. ACM, 153–164. <https://doi.org/10.1145/1989284.1989302>
- [5] Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider (Eds.). 2003. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press.
- [6] Eftychia Baikousi, Georgios Rogkakos, and Panos Vassiliadis. 2011. Similarity measures for multidimensional data. In *Proceedings of the 27th International Conference on Data Engineering, ICDE 2011, April 11-16, 2011, Hannover, Germany*, Serge Abiteboul, Klemens Böhm, Christoph Koch, and Kian-Lee Tan (Eds.). IEEE Computer Society, 171–182. <https://doi.org/10.1109/ICDE.2011.5767869>
- [7] Leopoldo E. Bertossi, Loreto Bravo, Enrico Franconi, and Andrei Lopatenko. 2005. Complexity and Approximation of Fixing Numerical Attributes in Databases Under Integrity Constraints. In *10th International Symposium on Database Programming Languages (DBPL) (LNCS, Vol. 3774)*. Springer, 262–278. https://doi.org/10.1007/11601524_17
- [8] Sandro Bimonte, Libo Ren, and Nestor Koueya. 2020. A linear programming-based framework for handling missing data in multi-granular data warehouses. *Data Knowl. Eng.* 128 (2020), 101832. <https://doi.org/10.1016/j.datak.2020.101832>
- [9] Philip Bohannon, Michael Flaster, Wenfei Fan, and Rajeev Rastogi. 2005. A Cost-Based Model and Effective Heuristic for Repairing Constraints by Value Modification. In *ACM SIGMOD International Conference on Management of Data*. ACM, 143–154. <https://doi.org/10.1145/1066157.1066175>
- [10] Peter Buneman, Sanjeev Khanna, and Wang-Chiew Tan. 2001. Why and Where: A Characterization of Data Provenance. In *8th International Conference on Database Theory (ICDT) (LNCS, Vol. 1973)*, Jan Van den Bussche and Victor Vianu (Eds.). Springer, 316–330. https://doi.org/10.1007/3-540-44503-X_20
- [11] Surajit Chaudhuri, Anish Das Sarma, Venkatesh Ganti, and Raghav Kaushik. 2007. Leveraging aggregate constraints for deduplication. In *ACM SIGMOD International Conference on Management of Data*. ACM, 437–448. <https://doi.org/10.1145/1247480.1247530>
- [12] Jan Chomicki. 2007. Consistent Query Answering: Five Easy Pieces. In *11th International Conference on Database Theory (ICDT) (LNCS, Vol. 4353)*. Springer, 1–17. https://doi.org/10.1007/11965893_1
- [13] Curtis E. Dyreson, Torben Bach Pedersen, and Christian S. Jensen. 2003. Incomplete Information in Multidimensional Databases. In *Multidimensional Databases: Problems and Solutions*, Maurizio Rafanelli (Ed.). Idea Group, 282–309. <https://doi.org/10.4018/978-1-59140-053-0.ch010>
- [14] Wenfei Fan and Floris Geerts. 2009. Relative information completeness. In *Proceedings of the Twenty-Eighth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS)*. ACM, 97–106. <https://doi.org/10.1145/1559795.1559811>
- [15] Floris Geerts, Giansalvatore Mecca, Paolo Papotti, and Donatello Santoro. 2013. The LLUNATIC Data-Cleaning Framework. *PVLDB* 6, 9 (2013), 625–636. <https://doi.org/10.14778/2536360.2536363>
- [16] Matteo Golfarelli and Elisa Turricchia. 2014. A characterization of hierarchical computable distance functions for data warehouse systems. *Decis. Support Syst.* 62 (2014), 144–157. <https://doi.org/10.1016/j.dss.2014.03.011>
- [17] Todd J. Green, Gregory Karvounarakis, and Val Tannen. 2007. Provenance semirings. In *ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS)*. ACM, 31–40. <https://doi.org/10.1145/1265530.1265535>
- [18] Tomasz Imielinski and Witold Lipski Jr. 1984. Incomplete Information in Relational Databases. *J. ACM* 31, 4 (1984), 761–791. <https://doi.org/10.1145/1634.1886>
- [19] Zachary G. Ives, Todd J. Green, Grigoris Karvounarakis, Nicholas E. Taylor, Val Tannen, Partha Pratim Talukdar, Marie Jacob, and Fernando C. N. Pereira. 2008. The ORCHESTRA Collaborative Data Sharing System. *SIGMOD Rec.* 37, 3 (2008), 26–32. <https://doi.org/10.1145/1462571.1462577>
- [20] Grigoris Karvounarakis, Zachary G. Ives, and Val Tannen. 2010. Querying data provenance. In *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2010, Indianapolis, Indiana, USA, June 6-10, 2010*, Ahmed K. Elmagarmid and Divyakant Agrawal (Eds.). ACM, 951–962. <https://doi.org/10.1145/1807167.1807269>
- [21] Chris Mayfield, Jennifer Neville, and Sunil Prabhakar. 2010. ERACER: a database approach for statistical inference and data cleaning. In *ACM SIGMOD International Conference on Management of Data*. ACM, 75–86. <https://doi.org/10.1145/1807167.1807178>
- [22] Robinson Meyer and Alexis C. Madrigal. 2021. Why the Pandemic Experts Failed. <https://www.theatlantic.com/science/archive/2021/03/americas-coronavirus-catastrophe-began-with-data/618287>
- [23] Arindam Nandi, Ying Yang, Oliver Kennedy, Boris Glavic, Ronny Fehling, Zhen Hua Liu, and Dieter Gawlick. 2016. Mimir: Bringing CTables into Practice. *CoRR* abs/1601.00073 (2016). arXiv:1601.00073 <http://arxiv.org/abs/1601.00073>
- [24] Lamia Oukid, Omar Boussaid, Nadja Benblidia, and Fadila Bentayeb. 2016. TLabel: A New OLAP Aggregation Operator in Text Cubes. *Int. J. Data Warehousing and Mining* 12, 4 (2016), 54–74. <https://doi.org/10.4018/IJDWM.2016100103>
- [25] Pierre Senellart, Louis Jachiet, Silviu Maniu, and Yann Ramusat. 2018. Provenance and Probability Management in PostgreSQL. *PVLDB* 11, 12 (2018), 2034–2037. <https://doi.org/10.14778/3229863.3236253>
- [26] Bartolomeo Stellato, Goran Banjac, Paul Goulart, Alberto Bemporad, and Stephen P. Boyd. 2020. OSQP: an operator splitting solver for quadratic programs. *Mathematical Programming Computation* 12, 4 (2020), 637–672. <https://doi.org/10.1007/s12532-020-00179-2>
- [27] Minghe Yu, Guoliang Li, Dong Deng, and Jianhua Feng. 2016. String similarity search and join: a survey. *Frontiers Comput. Sci.* 10, 3 (2016), 399–417. <https://doi.org/10.1007/s11704-015-5900-5>