Database Repair for Multivalued Dependencies

Sima Jamali
Simon Fraser University
Vancouver, Canada
sja88@sfu.ca

Babak Salimi
University of California San Diego
San Diego, United States
bsalimi@ucsd.edu

David Mitchell
Simon Fraser University
Vancouver, Canada
mitchell@cs.sfu.ca

***Abstract*—We describe a set of SAT solver benchmark instances (CNF formulas) obtained from the problem of finding a minimal repair for a relational database to satisfy a multivalued dependency constraint.**

***Index Terms*—MVD, SAT encoding, database repair**

I. Multivalued Dependency (MVD)

A MultiValued Dependency (MVD) for a relational database is a constraint between sets of attributes in a relation. An MVD holds when there are three attributes, say X,Y and Z, where for each value of Z there is a specific set of values for each of X and Y, but the values of X and Y are independent [1]. An MVD is a tuple-generating dependency, meaning that presence of certain tuples in the relation imply that other tuples must also be present. If a relation does not satisfy an MVD, the relation can be modified to satisfy the constraint (repaired) either by adding tuples or removing tuples. (An empty database satisfies an MVD constraint). For a database D that does not satisfy an MVD constraint, the minimal database repair problem is to find another database D^* at minimal distance from D that satisfies the MVD. As distance function we use the symmetric difference, i.e. $|D - D^*|$. Our encoding as SAT is a decision version of the MaxSAT encoding of [2].

II. SAT Encoding

If database D does not satisfy an MVD constraint ϕ, it can be repaired either by adding tuples or by removing tuples. Each CNF formula expresses, for a given database D and MVD constraint ϕ, the question: Is it possible to modify D so that it satisfies ϕ by adding at most i_1 tuples and/or removing at most i_2 tuples?

Consider database D with schema $(X, Y, W_1, \ldots W_n)$, where the MVD constraint is on X, Y and some W_i. Let the (finite) domains of X and Y be $Dom(X)$ and $Dom(Y)$. We treat the remaining attributes as one variable Z, with $Dom(Z) = \prod_{W_i \in W_1 \ldots W_n} Dom(W_i)$. Now define the database D* from D as:

$$D^*(\mathbf{X}_1, \mathbf{Y}_2, \mathbf{Z}) = D(\mathbf{X}_1, \mathbf{Y}_1, \mathbf{Z}) \wedge D(\mathbf{X}_2, \mathbf{Y}_2, \mathbf{Z})$$

D^* is the repaired version of database D that satisfies MVD by adding all the missing tuples.

To encode the problem, we associate to each tuple t in D^* (which includes all tuples of D) a variable V_t. Algorithm 1 generates the encoding (for the slightly simplified cases where $i_1 = i_2$). We express the MVD constraint as a set of clauses of the form $(V_{t_1} \wedge V_{t_2}) \longrightarrow V_{t_3}$, where t_1 and t_2 are tuples in D and t_3 is a tuple that must, as a consequence, be in D to satisfy ϕ. This is done by the second for loop in Algorithm 1. We also include clauses representing the fact that we are allowed to add up to i_1 clauses (those in D^* but not in D) or remove up to i_2 clauses. This is done by the first loop in Algorithm 1. It ensures that, for each subset $i_2 + 1$ existing tuples, not all can be removed, by including $\neg(\neg V_{t_1} \wedge \ldots \wedge \neg V_{t_{i+1}})$. A similar encoding ensures that, for each set of $i_1 + 1$ missing tuples (in D), not all can be added, by including $\neg(V_{t_1} \wedge \ldots \wedge V_{t_{i+1}})$.

Algorithm 1: Encodes problem of deciding D can be repaired to satisfy a MVD for with at most $i - 1$ deletions or $i - 1$ additions as a CNF formula.

Input: A database D with vairables $\mathbf{X} \cup \mathbf{Y} \cup \mathbf{Z}$
Output: A CNF Ψ
Compute $D^*(\mathbf{X}_1, \mathbf{Y}_2, \mathbf{Z}) = D(\mathbf{X}_1, \mathbf{Y}_1, \mathbf{Z}) \wedge D(\mathbf{X}_2, \mathbf{Y}_2, \mathbf{Z})$
for $t_1, \ldots t_i \in D^*$ **do**
  If $t_1, \ldots t_i \in D$, add a clause $(V_{t_1} \vee \ldots \vee V_{t_i})$ to Ψ
  If $t_1, \ldots t_i \in D^* - D$ add a clause $(\neg V_{t_1} \vee \ldots \vee \neg V_{t_i})$ to Ψ
Compute
$C(\mathbf{X}_1, \mathbf{Y}_1, \mathbf{X}_2, \mathbf{Y}_2, \mathbf{Z}) = D^*(\mathbf{X}_1, \mathbf{Y}_1, \mathbf{Z}) \wedge D^*(\mathbf{X}_2, \mathbf{Y}_2, \mathbf{Z})$
for $t \in C$ **do**
  $t_1 \leftarrow t[\mathbf{X}_1, \mathbf{Y}_1, \mathbf{Z}]$; $t_2 \leftarrow t[\mathbf{X}_2, \mathbf{Y}_2, \mathbf{Z}]$;
  $t_3 \leftarrow t[\mathbf{X}_1, \mathbf{Y}_2, \mathbf{Z}]$
  Add a clause $(\neg V_{t_1} \vee \neg V_{t_2} \vee V_{t_3})$ to Ψ

III. Instance Naming

The file names are of the form MVD-*database-sequential#*-i_1-i_2. Our instances submitted to the 2021 SAT Solver competition are generated from random subsets 300 to 600 rows of the database https://archive.ics.uci.edu/ml/datasets/adult. The resulting file names then are of the form MVD-ADS-*sample#*-i_1-i_2.

Funded in part by the Natural Sciences and Engineering Research Council of Canada (NSERC), through a Discovery Grant to the third author.

References

[1] S. Abiteboul, R. Hull, and V. Vianu, "Foundations of Databases,", January 1955.
[2] B. Salimi, L. Rodriguez, B. Howe and D. Suciu, "Interventional fairness: Causal database repair for algorithmic fairness," Proceedings of the 2019 International Conference on Management of Data, pp. 793-810, August 2019.