Discussion of:
A Unified Model for Data and Constraint Repair

Vincent J. Maccio

Department of Computing and Software
McMaster University
Hamilton, Ontario, Canada
Constraint Repair/Alteration - Questions

- Q: When would you do this, what's your justification?
- Q: Can you think of a specific example of such a case?
- Q: When would you be apprehensive about altering, or generating new constraints? Why?
• Norms change with time, poorly thought out constraints, merging constraints conflict, etc.
• (Address) → (Phone) Correct 25 years ago? Now?
• Q: When would you be apprehensive about altering, or generating new constraints? Why?
• Constraints represent rigid business rules. The constraints should describe the data, and not the other way around.
• Q: What’s the greatest strength and shortcoming of the model?
• Q: Why do data repairs cost \((1 + r)\) and not \(r\)?
• Q: Why are DLs measured in cells and not tuples?
Repair Model/Framework

- A: Clever cost function which allows for “better” data and constraint repairs. The constraints are limited to FDs.
- A?: Overhead cost. Distinguishes between “changing” a record to itself.
- A?: Discourage over-fitting.
Experimental Results

Fig. 1: Precision vs. error rate

Fig. 2: Recall vs. error rate

Fig. 3: Scalability wrt no. of tuples N

Fig. 4: Scalability wrt error rate \( e \)

Fig. 5: Scalability wrt no. of constraints

Fig. 6: Scalability wrt no. of attributes
Overall

- What was the most enjoyable part about the paper?
- What would you do differently if you were the author?