Tracing Data Errors Using View-Conditioned Causality

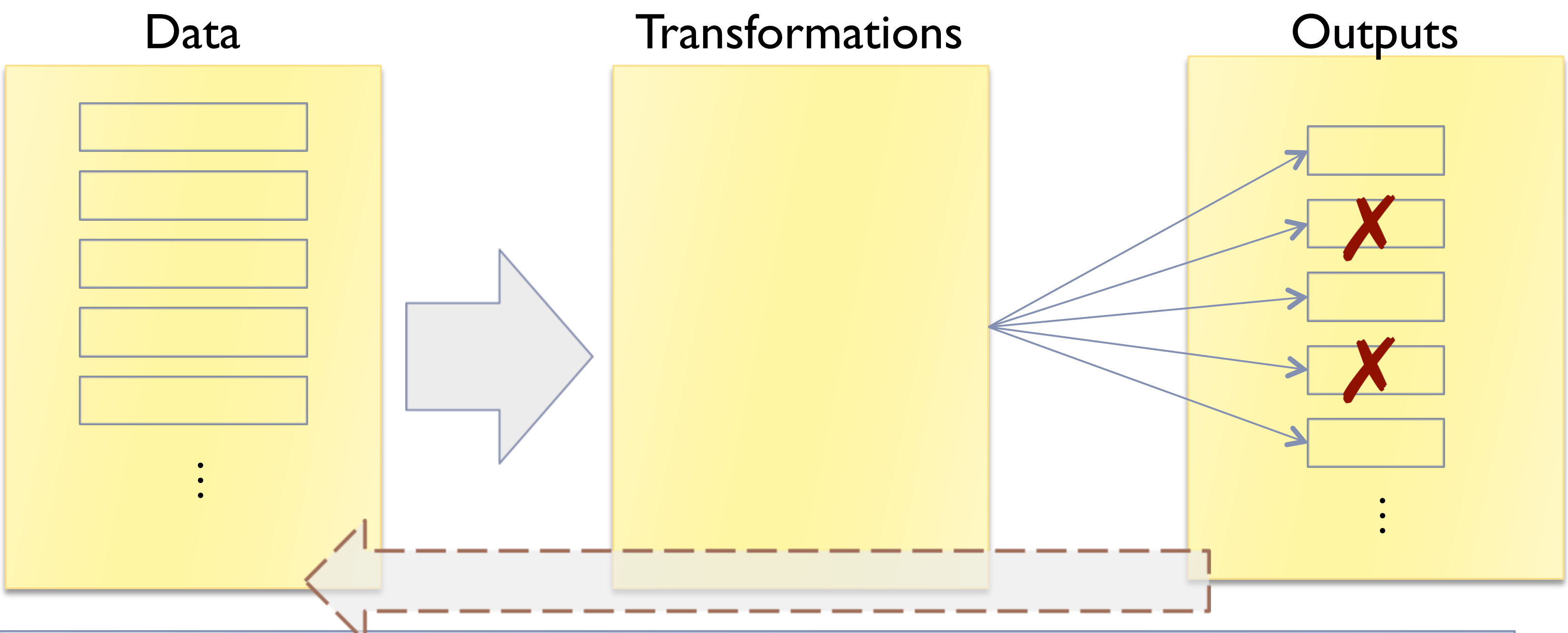
Alexandra Meliou*

with Wolfgang Gatterbauer*, Suman Nath§, and Dan Suciu*

**University of Washington, §Microsoft Research*

University of Washington

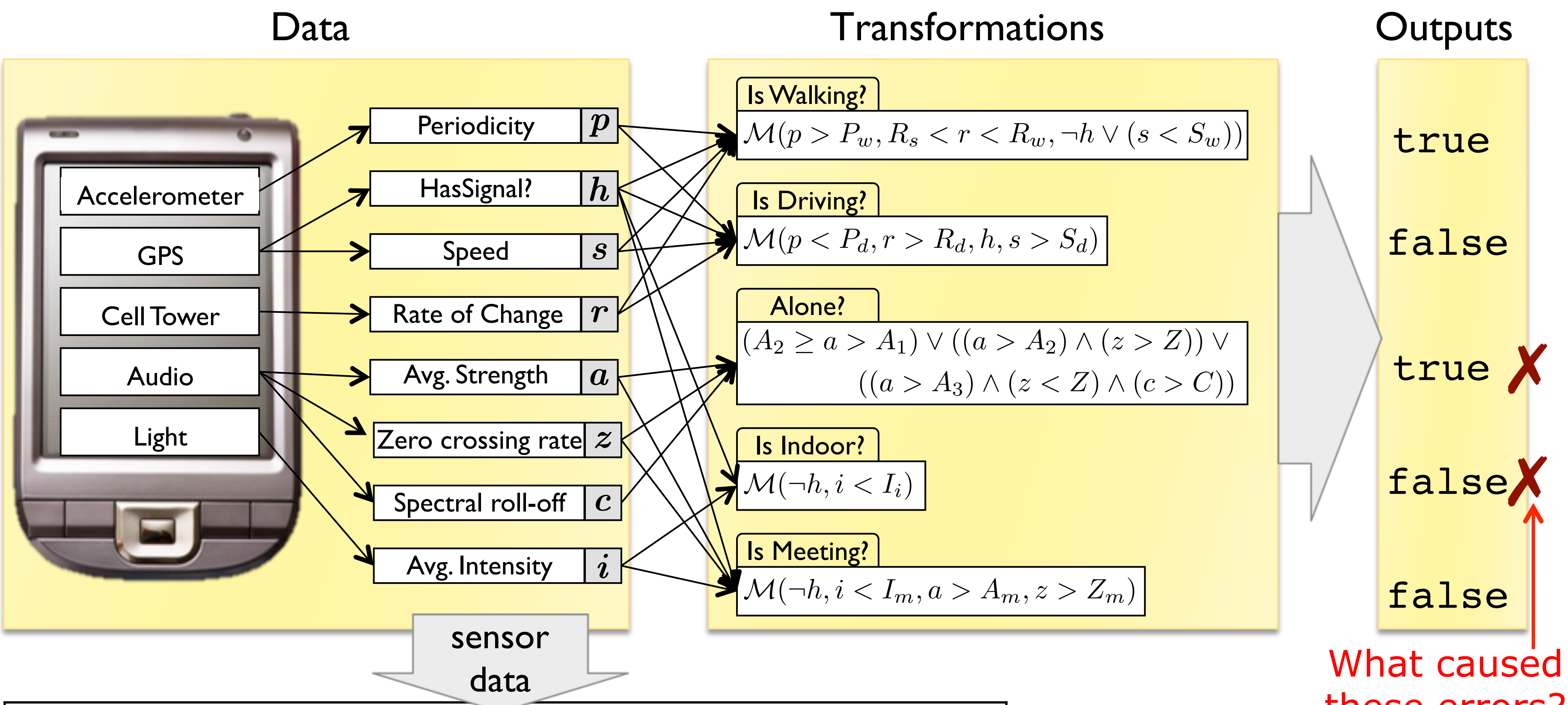
General Problem Setting



If one or more of the outputs are deemed erroneous, can we find the tuples in the base data responsible for that error?
Correcting those can fix even more potential errors in the output.

Provenance helps narrow down the candidate tuples in the input data. The challenge is to identify the input tuples that can best explain the observed errors.

Focus: Context Aware Recommendations



0.016	True	0.067	0	0.4	0.004	0.86	0.036	10
0.0009	False	0	0	0.2	0.0039	0.81	0.034	68
0.005	True	0.19	0	0.03	0.003	0.75	0.033	17
0.0008	True	0.003	0	0.1	0.003	0.8	0.038	18

Sensors may be faulty or inhibited

It is not straightforward to spot such errors in the provenance

Contributions

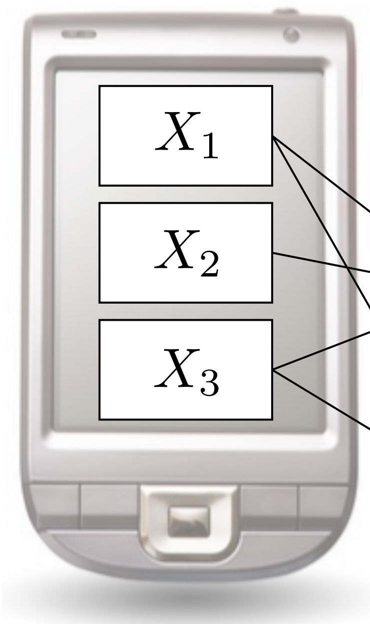
- Introduce view-conditioned causality and responsibility for tracing errors in views and transformations to source data
  - The presence of errors is often obvious in the transformations but not the source data (*post-factum cleaning*)
- Non-trivial reduction of causality and responsibility computation to a satisfiability problem
- An optimized conversion algorithm that reduces the SAT problem size
- Illustration of effectiveness in a real-world classifier-based recommendation system using mobile sensor data
  - High average precision, and almost 90% correction ratio in some cases

Running Example

Input variables can be from a continuous or discrete domain

Example:

Input			Results in output
X_1	$=$	3	$Z_1 = \texttt{true}$
X_2	$=$	$\texttt{true}$	$Z_2 = \underline{\texttt{true}}$
X_3	$=$	4	



Φ_1

$$Z_1 = (X_1 < 5) \wedge (X_3 = 4) \vee \neg X_2$$

Φ_2

$$Z_2 = (X_1 > 2) \wedge (X_3 \geq 4)$$

error

But what if we know that the first classifier should evaluate to true, and the second to false?

Ground truth: $\hat{Z}_1 = \texttt{true}, \hat{Z}_2 = \underline{\texttt{false}}$

View-Conditioned Causality

Refer to the paper for the formal definitions

- A set of input variables is a counterfactual cause, if changing their values results in the correct output for all transformations, and the set is minimal.

Example:

Evaluate to:

$$Z_1 = (X_1 < 5) \wedge (X_3 = 4) \vee \neg X_2 = \texttt{true}$$
$$Z_2 = (X_1 > 2) \wedge (X_3 \geq 4) = \texttt{true}$$

Ground truth:

$$\hat{Z}_1 = \texttt{true}$$
$$\hat{Z}_2 = \texttt{false}$$

$$X_1 = 3$$
$$X_2 = \texttt{true}$$
$$X_3 = 4$$

Counterfactual causes:		Change:	Gives output:	
$\{X_1\}$	changing values $\Rightarrow$	$X_1' = 1$	$\{Z_1', Z_2'\} = \{\hat{Z}_1, \hat{Z}_2\}$	ground truth
$\{X_2, X_3\}$	$\Rightarrow$	$\{X_2', X_3'\} = \{\texttt{false}, 2\}$	$\{Z_1', Z_2'\} = \{\hat{Z}_1, \hat{Z}_2\}$	

View-Conditioned Causality

Refer to the paper for the formal definitions

- A variable is a cause if it is a part of a counterfactual cause
- If $X_i \cup \Gamma$ is a counterfactual cause, Γ is a contingency for X_i

Responsibility: $\rho_{X_i} = \dfrac{1}{1 + min_\Gamma |\Gamma|}$

The smaller the contingency set, the higher the responsibility

Example:

Evaluate to:

$$Z_1 = (X_1 < 5) \wedge (X_3 = 4) \vee \neg X_2 = \texttt{true}$$
$$Z_2 = (X_1 > 2) \wedge (X_3 \geq 4) = \texttt{true}$$

Ground truth:
$\hat{Z}_1 = \texttt{true}$
$\hat{Z}_2 = \texttt{false}$

$X_1 = 3$
$X_2 = \texttt{true}$
$X_3 = 4$

Counterfactual causes:	Change:	Responsibility:
$\{X_1\}$	$X_1' = 1$	$\rho_{X_1} = 1$
$\{X_2, X_3\}$	$\{X_2', X_3'\} = \{\texttt{false}, 2\}$	$\rho_{X_2} = \rho_{X_3} = \frac{1}{2}$

causes

Our Approach to Post-Factum Cleaning

- Compute all causes and rank them by their responsibility.
  - Use the ranking as an indicator for error tracing

- But: Computing responsibility is hard for general Boolean formulas [Eiter et al. 2002], and even for conjunctive queries [PVLDB 2010]

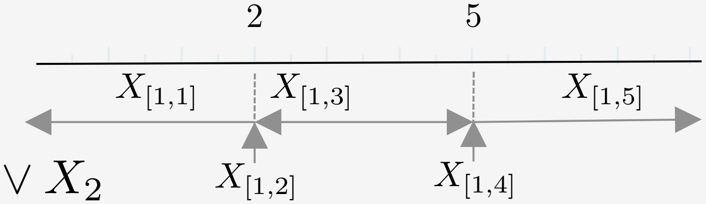
- Transform causality into a satisfiability problem and use highly optimized SAT solvers, which are very efficient in practice
  - We explain how we do this in 4 main steps

Reduction to SAT

1. **Map continuous input to Boolean partition variables**

Example (cont.):

$$\begin{aligned}\Phi_1 : Z_1 &= (\underline{X_1 < 5}) \wedge (X_3 = 4) \vee \neg X_2 \\ &= (X_{[1,1]} \vee X_{[1,2]} \vee X_{[1,3]}) \wedge X_{[3,2]} \vee X_2 \\ \Phi_2 : Z_2 &= (\underline{X_1 > 2}) \wedge (X_3 \geq 4) \\ &= (X_{[1,3]} \vee X_{[1,4]} \vee X_{[1,5]}) \wedge (X_{[3,2]} \vee X_{[3,3]})\end{aligned}$$



Non-overlapping intervals for X_1

2. **When the intervals are non-overlapping, we can easily model their correlation with a constraint**

$$\Psi_i = \underbrace{\left(\bigvee_j X_{[i,j]}\right)}\underbrace{\left(\bigwedge_{j<l} (\neg X_{[i,j]} \vee \neg X_{[i,l]})\right)} \qquad \Psi = \bigwedge \Psi_i$$

At least one is true + No two are true together = exactly one is true

Example (cont.): $\Psi_3 = (X_{[3,1]} \vee X_{[3,2]} \vee X_{[3,3]})$

$$\wedge(\neg X_{[3,1]} \vee \neg X_{[3,2]}) \wedge (\neg X_{[3,1]} \vee \neg X_{[3,3]}) \wedge (\neg X_{[3,2]} \vee \neg X_{[3,3]})$$

Reduction to SAT

Running Example:

$$\Phi_1 \quad : Z_1 \quad = \quad (X_1 < 5) \wedge (X_3 = 4) \vee \neg X_2$$
$$\Phi_2 \quad : Z_2 \quad = \quad (X_1 > 2) \wedge (X_3 \geq 4)$$

Ground truth:
$\hat{Z}_1 = \texttt{true}$
$\hat{Z}_2 = \texttt{false}$

Input values:
$X_1 = 3$
$X_2 = \texttt{true}$
$X_3 = 4$

3. a. **Construct a Boolean formula whose satisfying assignments produce the correct output**

$$\hat{\Phi} = \left(\bigwedge_{i.\hat{z}_i = \mathtt{T}} \Phi_i \right) \wedge \left(\bigwedge_{i.\hat{z}_i = \mathtt{F}} \neg \Phi_i \right)$$

Example (cont.): $\hat{\Phi} = \Phi_1 \wedge \neg \Phi_2$

All satisfying assignments of $\hat{\Phi}$ cause each Φ_i to evaluate to its ground truth

b. **Construct a Boolean formula whose satisfying assignments satisfy $\hat{\Phi}$, and also change the value of X_i**

$$\Phi_{\mathrm{SAT}} = \neg \hat{\Phi} \left[\theta(\boldsymbol{X}_{[i]}) \right] \wedge \hat{\Phi} \left[\neg \theta(X_{[i,j]}) \right] \wedge \Psi \left[\neg \theta(X_{[i,j]}) \right]$$ (hard constraint)

Example (cont.): X_1 is a cause iff the following formula is satisfiable:

$$\Phi_{SAT} = \neg \hat{\Phi}[\underline{\{X_{[1,1]}, X_{[1,2]}, X_{[1,3]}, X_{[1,4]}, X_{[1,5]}\}} = \{\mathtt{F}, \mathtt{F}, \mathtt{T}, \mathtt{F}, \mathtt{F}\}]$$
$$\wedge \hat{\Phi}[X_{[1,3]} = \mathtt{F}] \wedge \Psi[X_{[1,3]} = \mathtt{F}]$$

Current assignment of X_1

Negate current assignment of X_1

Computing Responsibility with MaxSAT

Running Example:

$$\Phi_1 \quad : Z_1 \quad = \quad (X_1 < 5) \wedge (X_3 = 4) \vee \neg X_2$$
$$\Phi_2 \quad : Z_2 \quad = \quad (X_1 > 2) \wedge (X_3 \geq 4)$$

Ground truth:
$\hat{Z}_1 = \texttt{true}$
$\hat{Z}_2 = \texttt{false}$

Input values:
$X_1 = 3$
$X_2 = \texttt{true}$
$X_3 = 4$

4. **Construct "soft" constraints to find minimum contingency set**

$$\Phi_\theta = \bigwedge_{\theta(X_{[i,j]})=\texttt{T}} X_{[i,j]} \bigwedge_{\theta(X_{[i,j]})=\texttt{F}} \neg X_{[i,j]}$$

(soft constraint)

Example (cont.):

$$\Phi_\theta = \neg X_{[1,1]} \wedge \neg X_{[1,2]} \wedge X_{[1,3]} \wedge \neg X_{[1,4]} \wedge \neg X_{[1,5]} \wedge X_2 \wedge \neg X_{[3,1]} \wedge \neg X_{[3,2]} \wedge X_{[3,3]}$$

A partial MaxSAT solver tries to satisfy as many conjuncts of the soft constraint as possible, and thus produces an assignment as similar to the given one as possible

→ Minimum contingency

Experimental Setup

- Three individuals using our context-aware recommendation system on their mobile devices over a period of 3 weeks
- Dataset:
  - 800 different instances of user activity
  - 150 total hours of data during the 3 weeks
- The users recorded erroneous outputs, as well as whether sensors happened to be inhibited
- SAT reduction implemented in Java, output exported in standard DIMACS CNF and WCFN formats
- MiniSat (http://minisat.se/) and MiniMaxSat ([Heras et al. 2008]) solvers

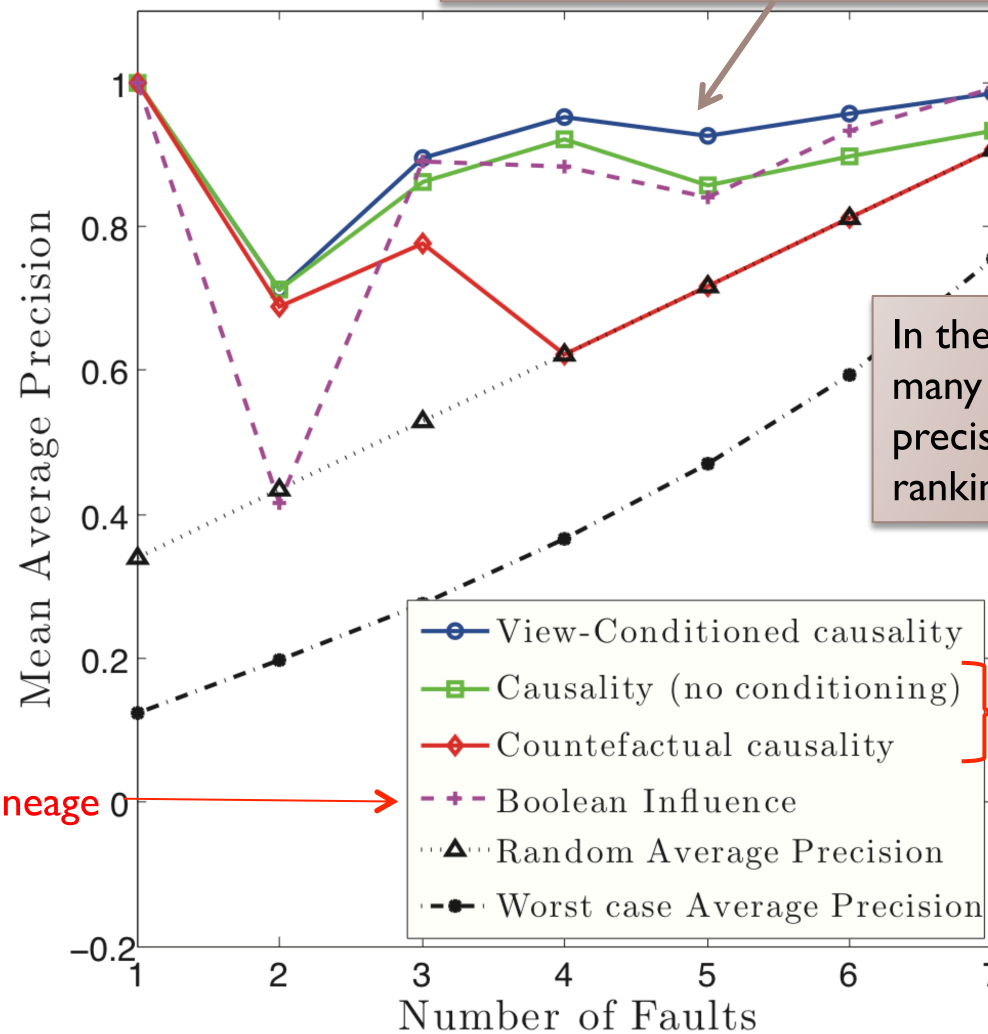
Average Precision

View-Conditioned causality produces more accurate error rankings than other approaches

800 different instances
5 sensory inputs
8 extracted features (variables)
3 users

Average precision is a metric of quality of a ranking.

If all erroneous variables are ranked first, then average precision is 1.



In the presence of many errors the avg precision of all rankings increases

Simpler causality schemes

Static analysis of lineage

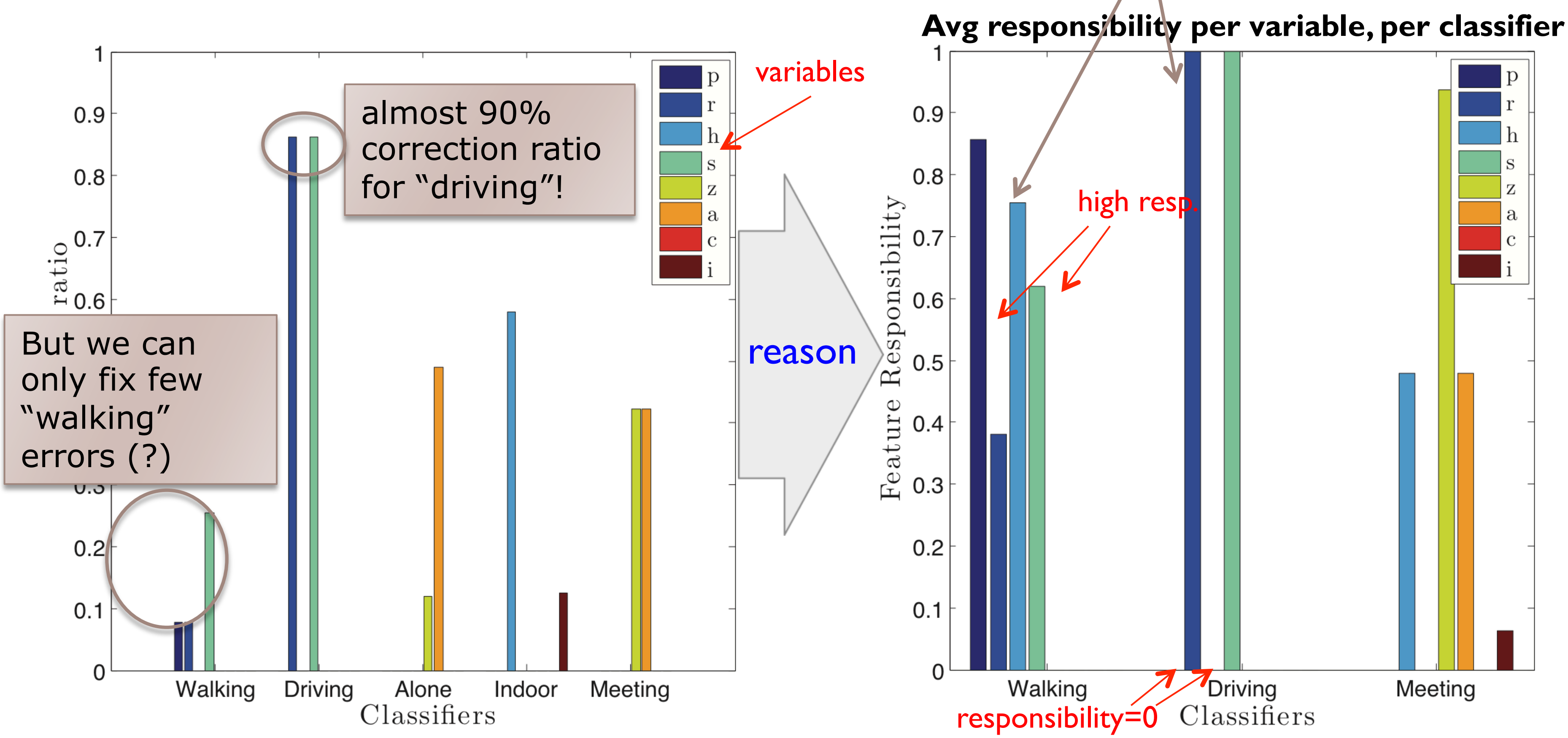
http://db.cs.washington.edu/causality/

Corrections

We select the highest responsibility variable, remove it from the evaluation of all classifiers, and record the portion of errors that get corrected per classifier

Driving has reliable features (low responsibility), means they are almost never causes of error

Walking has no reliable features

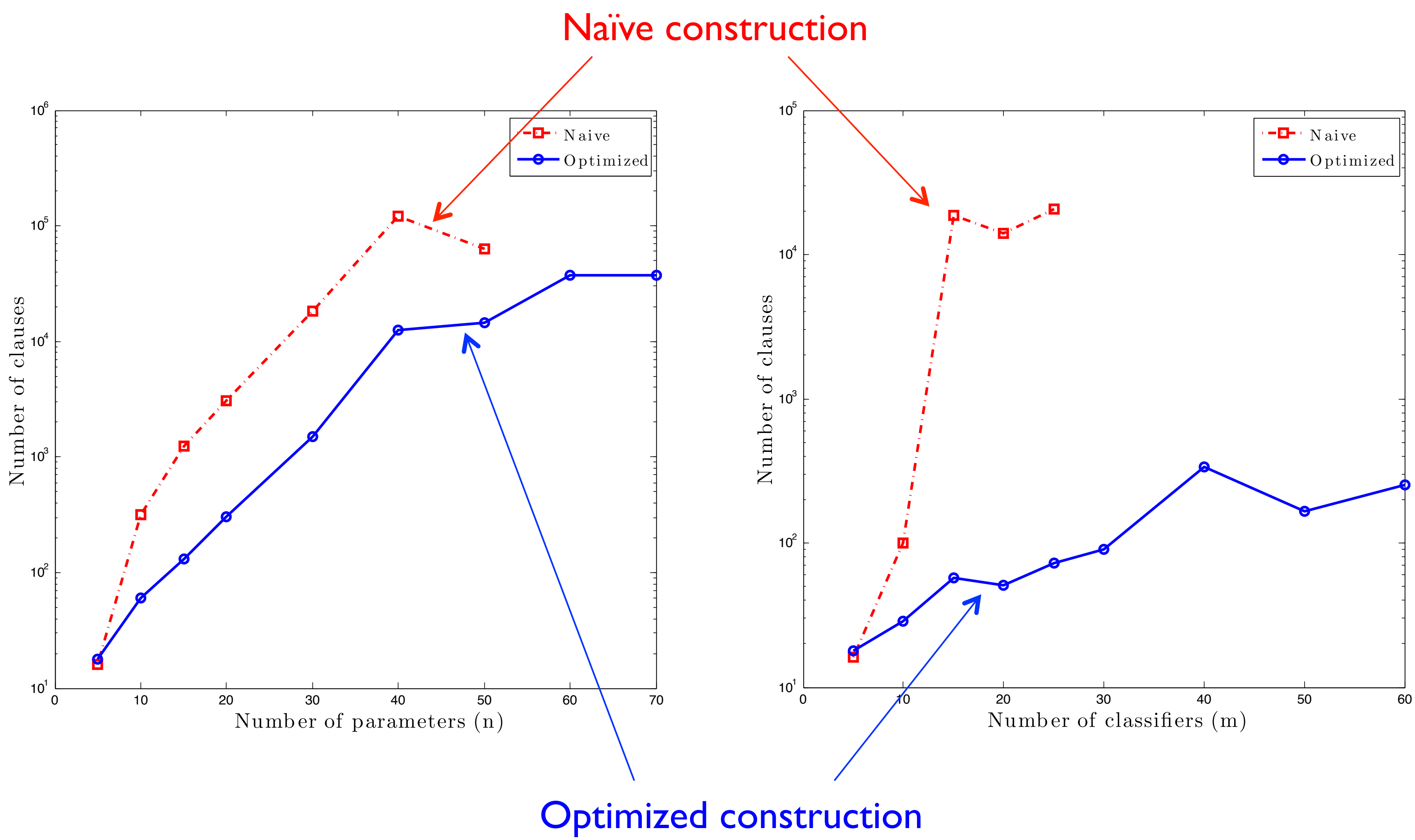


Conclusions

- Defined view-conditioned causality (VCC) and demonstrated its effectiveness in post-factum cleaning
  - Results show that VCC successfully identifies causes of error
- Described a non-trivial reduction to a satisfiability problem
- Also in the paper
  - Optimization of formula size (we achieve orders of magnitude improvement)
  - Scalability experiments
- Questions?

Additional Graphs

Improving the CNF Size



SAT Solver Runtime