Semantic Anomaly Detection in Online Data Sources

Orna Raz

Mary Shaw

Philip Koopman

Carnegie Mellon University

This research was supported by the National Science Foundation under Grant ITR-0086003

and by the Sloan Software Industry Center at Carnegie Mellon University

How it is today

The Register, March 19, 2001:

```
Reader Toby Doig nearly had a heart attack this morning
when he visited Datek to check out his share portfolio
and found that the Dow Jones Industrial Average (DJIA)
had taken the full brunt of the stock market wobble and
slumped just over 10,000 points to stand at 0.20
(down 99.999 percent, roughly)
...clearly there had been a terrible computing error...
```


With anomaly detection

- Toby would be notified the data feed is behaving suspiciously
- Our results support semantic anomaly detection
  - by inferring invariants about normal behavior
  - that serve as proxies for missing specifications

Outline

- *Setting:*

  dynamic data feeds, incomplete specifications

- Approach:

  inferring invariants using multiple existing techniques

- Feasibility results:

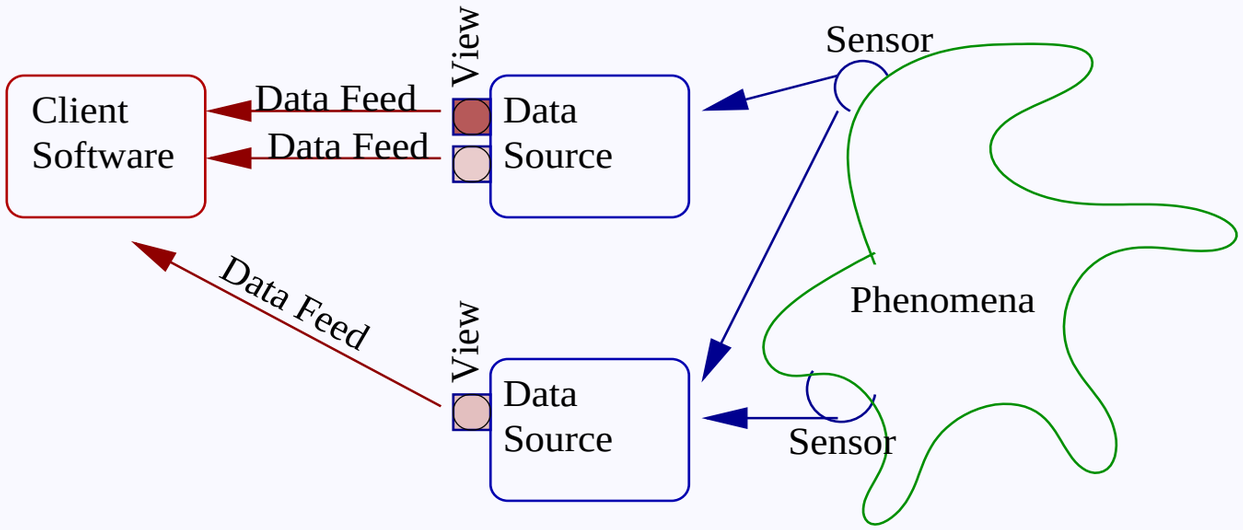
  effective detection on stock quote data feeds, using 2 techniques

Everyday software

Software people use for everyday purposes

- Not mission critical
- Must be dependable enough for practical use
  - Limited by dependability of elements
- Often incorporates elements maintained and developed by others
  - COTS (Commercial-Off-The-Shelf) components
  - Databases
  - Dynamic data feeds from online data sources

Dynamic data feeds



- A data source samples phenomena
  - Stock quotes, weather forecasts, airline ticket prices
- A data feed captures a particular usage of a data source via a view
  - KM (Kmart) quotes, Pittsburgh forecast, Pit–Tlv airfare

Dependability of data feeds

- Value increases by timely availability
- Evaluation/enhancement difficult
  - A priori validation insufficient
    - Data feed may change when being used
  - On-going failure detection and masking require specifications
    - Specifications are sketchy and incomplete
    - Free stock quote data feeds: at least 20min delay
- Semantic failures challenging
  - Results timely, well-formed BUT out-of-range, incorrect, inconsistent
- Availability under a semantic fault model needs to increase
  - Readiness for usage indicated by delivery of reasonable results

Goal

Long term: Increase the semantic availability of dynamic data feeds with incomplete specifications

This work: A first step

Results (this work):

- Can infer useful invariants about normal behavior of a data feed
  - ⇨ By using/adapting statistical and machine learning techniques
- Can use as proxies for missing specifications
  - ⇨ To effectively detect semantic anomalies
- Can do automatically, to a large extent

Outline

- Setting:

  dynamic data feeds, incomplete specifications

- *Approach:*
  - infer invariants using multiple existing techniques
  - use as proxies for missing specifications for anomaly detection
- Feasibility results:

  effective detection on stock quote data feeds, using 2 techniques

Goal (this work)

Given a sequence of observations

(time-ordered, with numeric attributes)

Build a (statistical) model to detect anomalies (outliers),

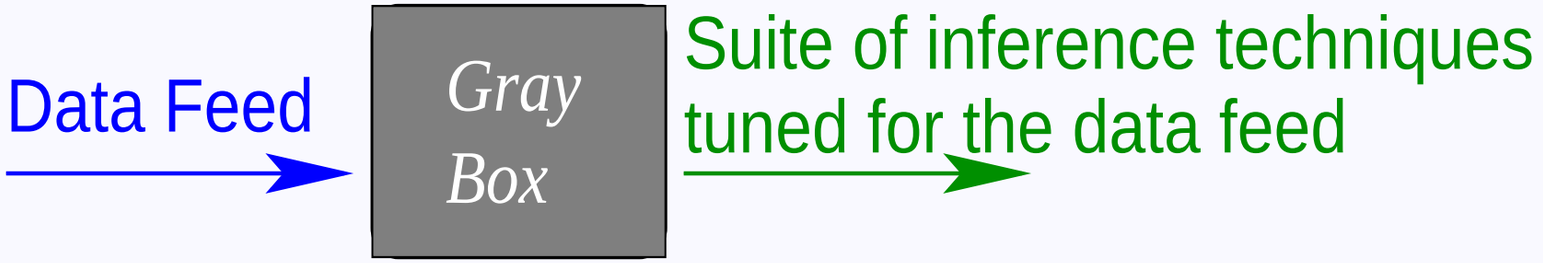
where a model is a set of invariants

...for example...

- ⟨attribute A value⟩ within 2 standard deviations of estimated mean
- ⟨attribute A value⟩ $<$ ⟨attribute B value⟩

Invariant inference framework

- Setup (human intervention required)



- Usage (human intervention permitted but not required)

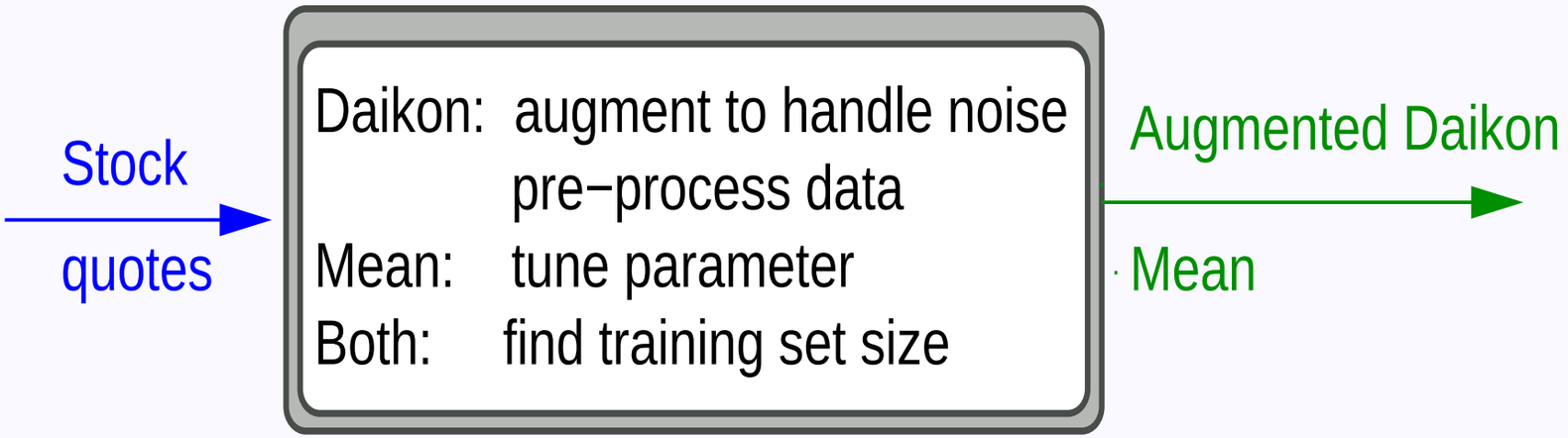
Example

- Data feed: Datek stock quote for `KM`. ⟨`cur`, `dhigh`⟩

time stamp	cur	dhigh
1/10/2000 9:00am	34.5	49.8
1/10/2000 9:10am	29.1	49.8
...	...	
1/10/2000 3:50pm	50.2	50.2
1/10/2000 4:00pm	46.3	50.2

- Techniques
  - Daikon [Ernst]: dynamically discover likely program invariants
  - Mean: estimate confidence interval for mean of attribute distribution

Setup



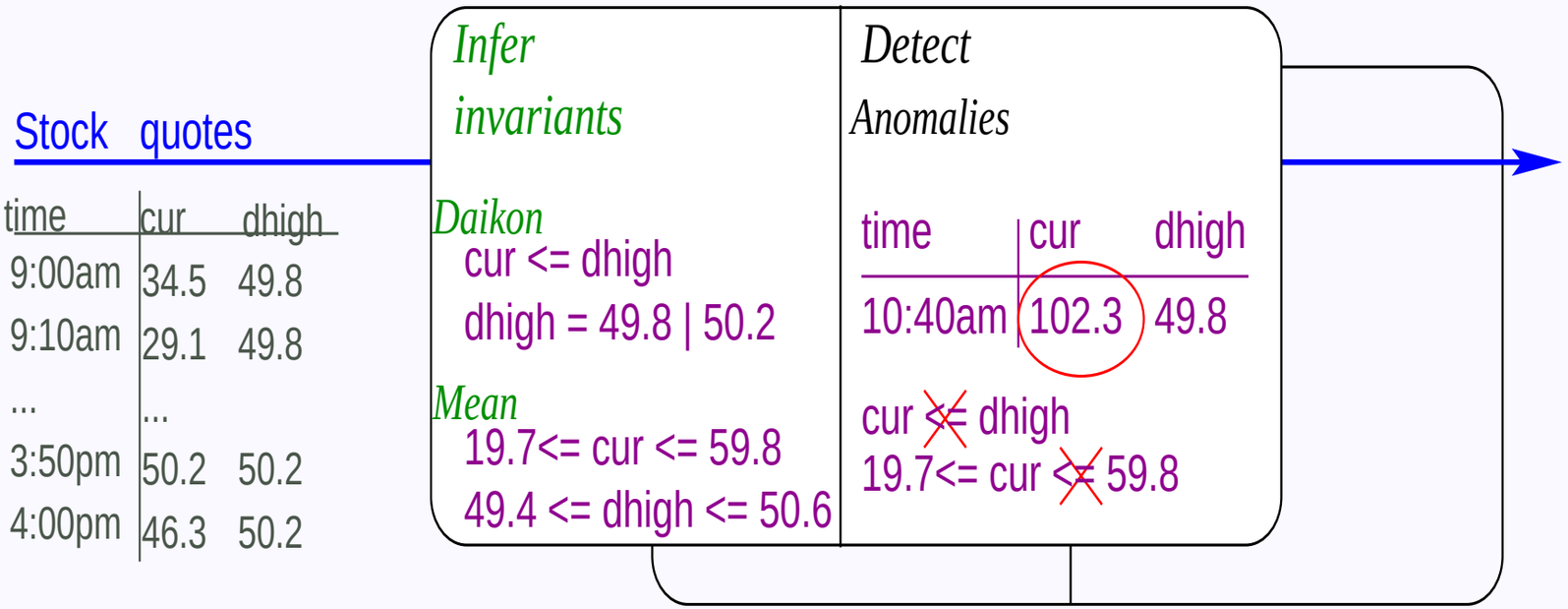
Usage

① Infer invariants

- Infer invariants over a moving window of observations

② Detect anomalies

- Evaluate each invariant over fresh observations
- Report when false



Appropriate for

- Everyday usage (non-critical), incomplete specifications
- Dynamic data, truth unknown before data used
- Ordered observations, numeric attributes
- Failures that result from
  - Faulty sensors
  - Human error when entering information
- Single feed, multiple feeds with a semantic relation
  - Redundancy stock quotes: Datek, Yahoo!Finance
  - Correlation rainfall amounts, river level

Outline

- Setting:

  dynamic data feeds, incomplete specifications

- Approach:

  inferring invariants using multiple existing techniques

- *Feasibility results:*

  effective detection on stock quote data feeds, using 2 techniques

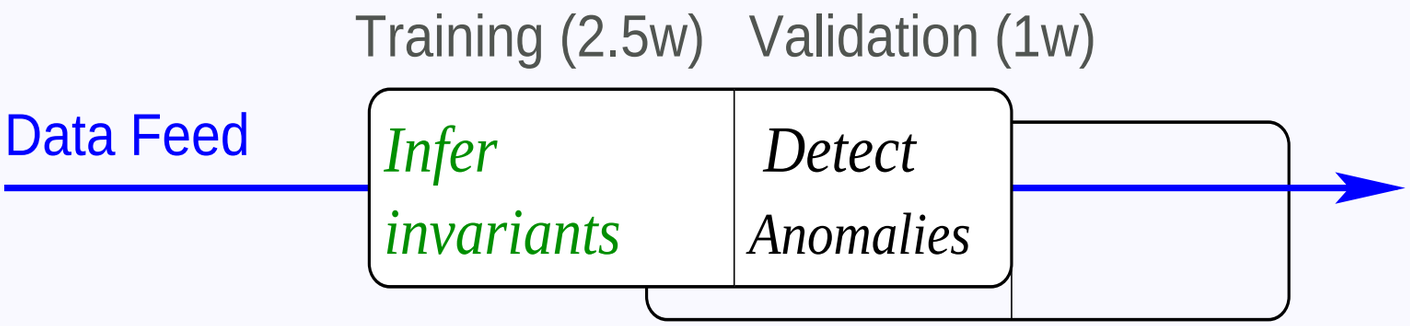
Data and methodology

- Techniques: Daikon, Mean
- Data: 3 stock-quote data sources; each viewed for 3 stock tickers

① Infer invariants over a training set

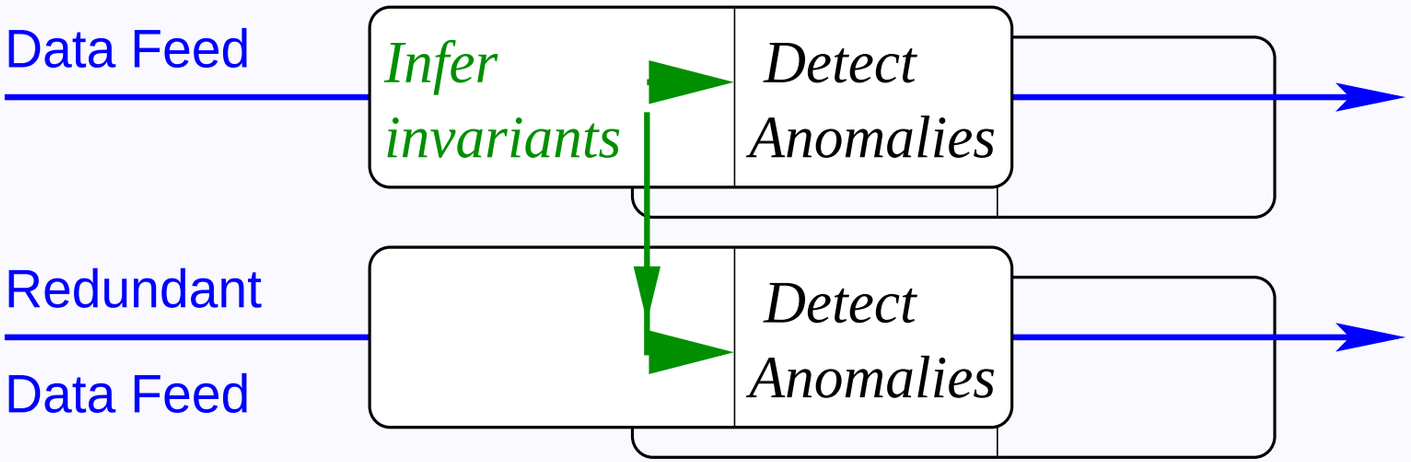
② Detect anomalies (evaluate invariants) over a disjoint validation set

- With and without a voting heuristic



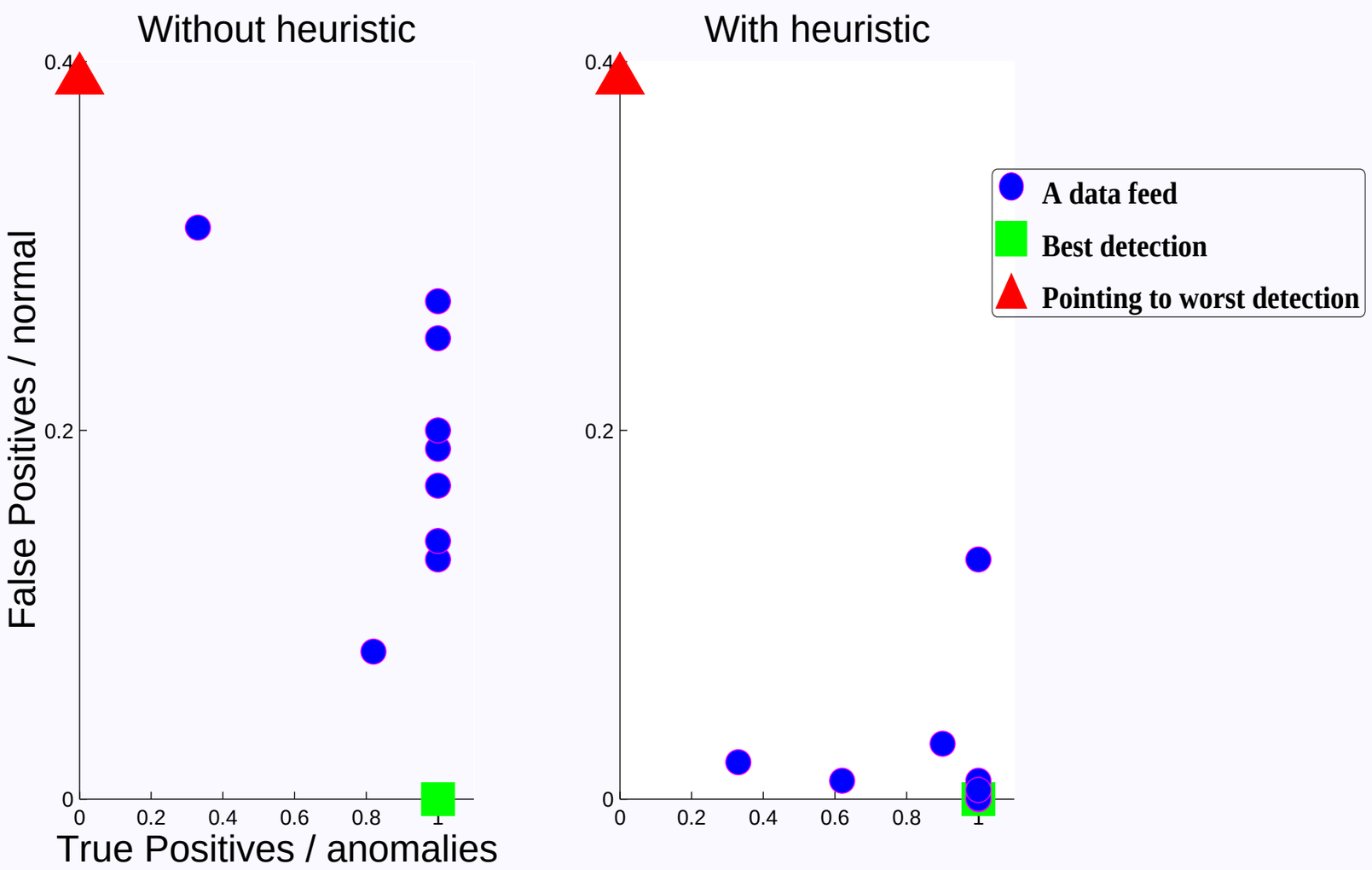
Voting heuristic

- Cross checks data with a redundant data feed
  - Multi version comparison; independence required
- Eliminate false positives: invariant breaks over both
- Identify false negatives: invariant breaks only over redundant



Results (example)

...Daikon...



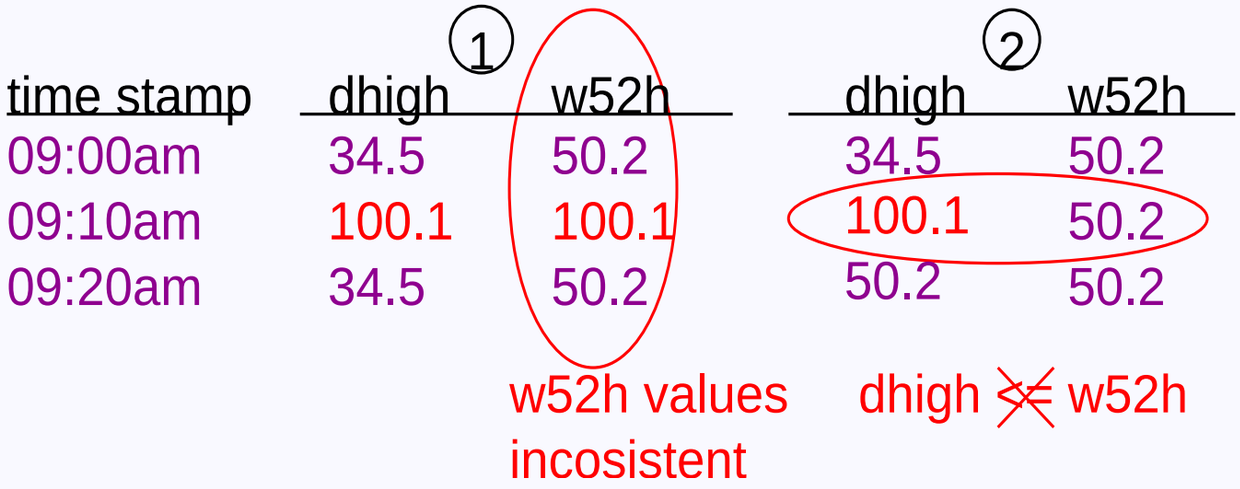
Results

- Misclassification $(\frac{\text{False positives}+\text{False negatives}}{\text{anomalies}+\text{normal}})$
  - $<$30% (due to 1 to 2 invariants; remove manually during setup)
  - $<$15%, usually $<$2% with voting heuristic
- Daikon and Mean complementary
- Detected anomalies helped us expose implicit specs of data source

Examples of implicit spec

① DS2 immediately updates w52l,w52h when exceeded by dlow,dhigh

② DS1 does not



- DS2 updates w52l, w52h infrequently if not exceeded by dlow, dhigh
- DS0 updates w52l, w52h frequently (about once a week)
- DS1, DS2 calculate volume differently for a particular stock

Feasibility demonstrated

...in the context of stock market tickers...

...for a single data feed with numeric attributes...

A first step in increasing the semantic availability of dynamic data feeds with incomplete specifications

...because raises a flag that will trigger repair...

- Can infer useful invariants about normal behavior of a data feed
- Can use as proxies for missing specifications
  - To effectively detect semantic anomalies
- Can do automatically, to a large extent
  - Training set size; parameter (for Mean)

Ongoing work

- Level of automation
  - Training set size; handling normal changes in data (concept drift)
  - Adjusting parameters of techniques
  - Attribute selection
- Generality
  - Multiple data feeds
  - Existing specifications
  - Techniques
  - Types of data feeds (categorical, text is beyond our scope)

Questions