Learning Error Patterns from Diagnosis Trouble Codes

Stefan Kriebel, Evgeny Kusmenko, Bernhard Rumpe, Igor Shumeiko

http://www.se-rwth.de/

SE Software Engineering | RWTH AACHEN UNIVERSITY

Motivation

- Large fleets produce massive amounts of self-diagnosis data consisting of diagnosis trouble codes (DTCs)
- DTCs are low level symptoms → an isolated DTC doesn‘t tell us much about the cause
- DTC combinations can indicate what is broken

- Our goal: Fully automated framework for error pattern detection

Modeling the domain

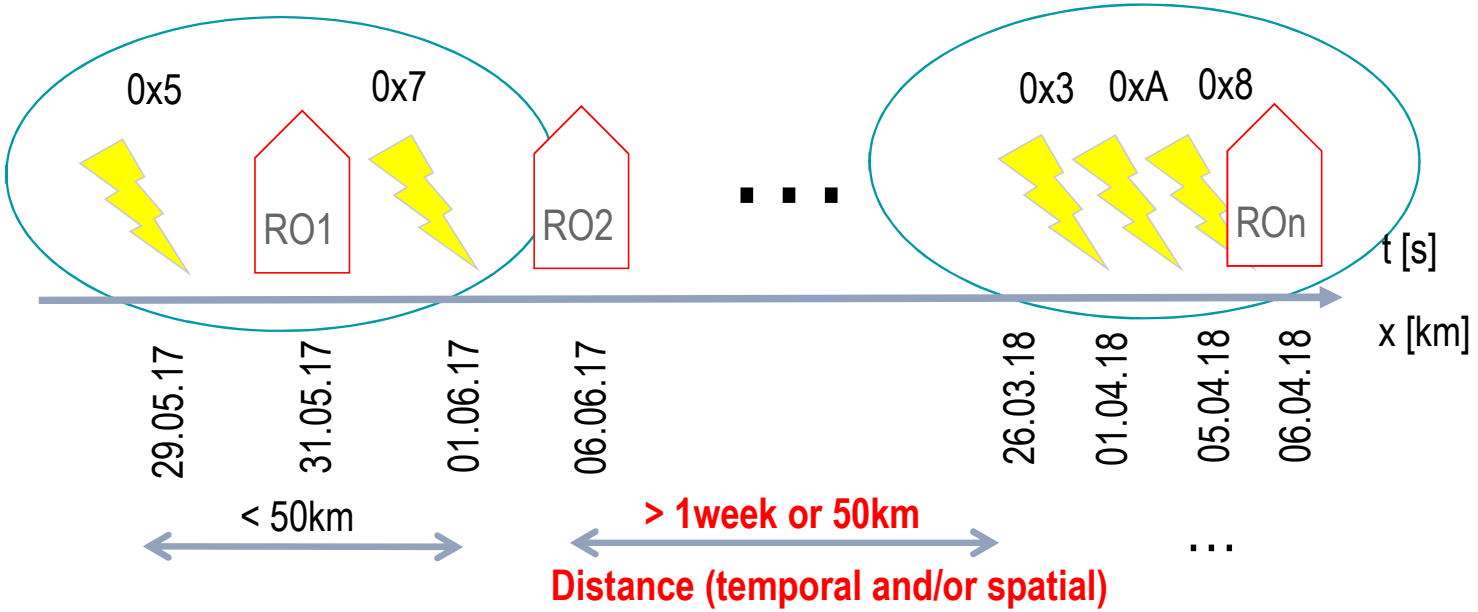
- How to model a vehicle's on-board diagnosis data for error pattern learning?
- Explicit representation in a vector space is infeasible
  - Too many dimensions
  - Extremely sparse vectors
- Instead:
  - Modeling a vehicle as a set of DTCs
  - Defining a similarity (or distance) function on vehicles
  - Using clustering algorithms which do not require a high dimensional embedding
- Example:
  - $v_1 = \{0x111222, 0x333222\}$,
  - $v_2 = \{0x111222, 0x444222\}$,
  - $\frac{|v_1 \cap v_2|}{|v_1 \cup v_2|} = 1/3$

In complex settings, a DTC can become an object carrying environmental conditions, sensor data, etc. Then more complex similarity functions are appropriate.

Jaccard similarity

Pre-processing: Case Clustering

- Data inside one car may come from different problems
- Single linkage clustering on each car
  - leads to a chaining effect of DTCs
  - Distance measure is distance traveled or time elapsed
  - Max distance is a crucial hyperparameter

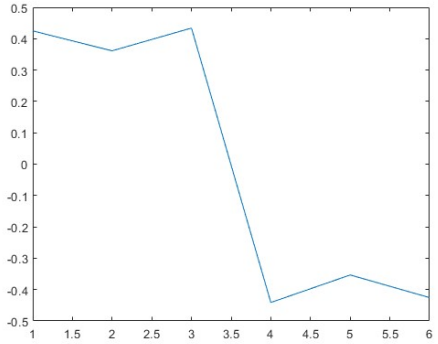


Spectral Clustering

- Well-established algebraic algorithm
- Takes similarity matrix as input

$$A=\begin{pmatrix} 1 & 0,9 & 0,8 & 0 & 0 & 0 \\ 0,9 & 1 & 0,6 & 0 & 0,2 & 0 \\ 0,8 & 0,6 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0,5 & 0,5 \\ 0 & 0,2 & 0 & 0,5 & 1 & 0,8 \\ 0 & 0 & 0 & 0,5 & 0,8 & 1 \end{pmatrix}, \quad L=D-A,\ D=\mathrm{diag}(1^{T} A)$$

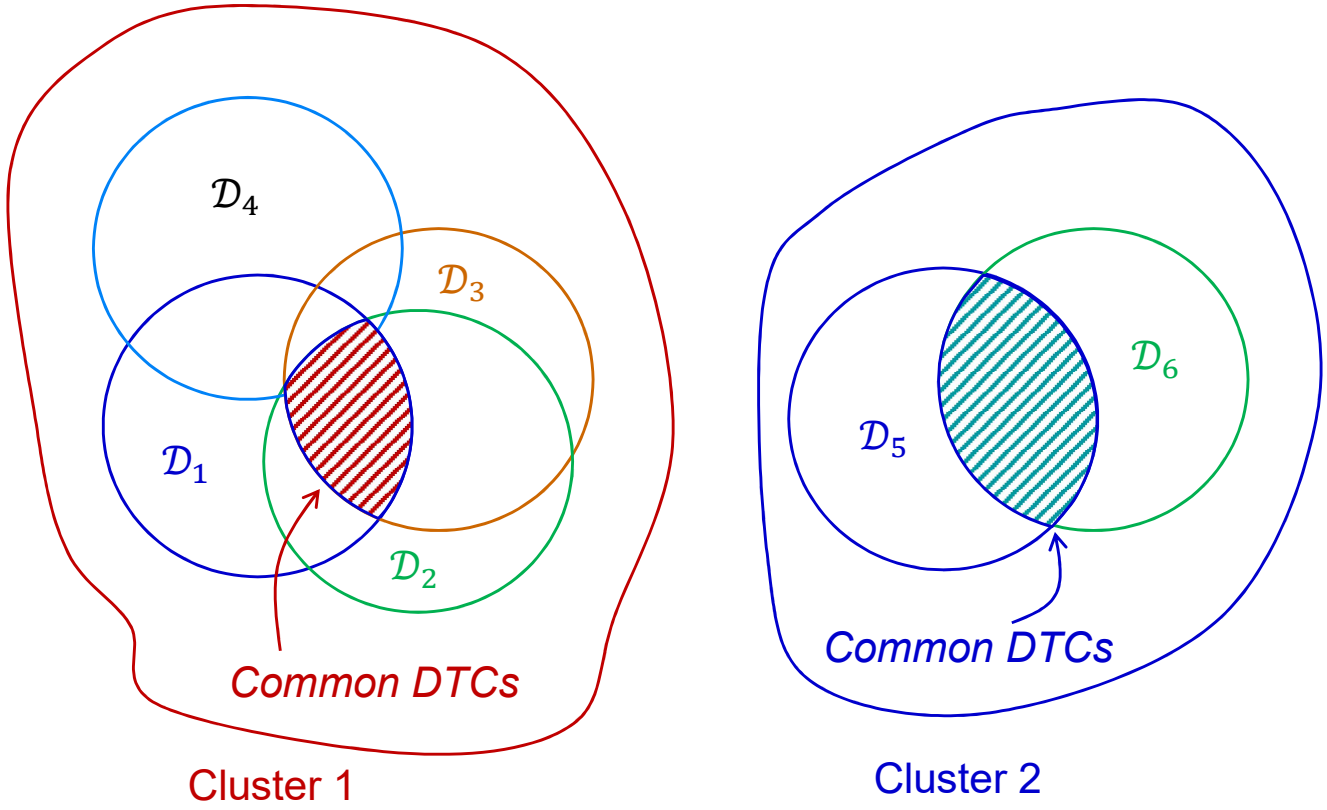
2nd eigvec =



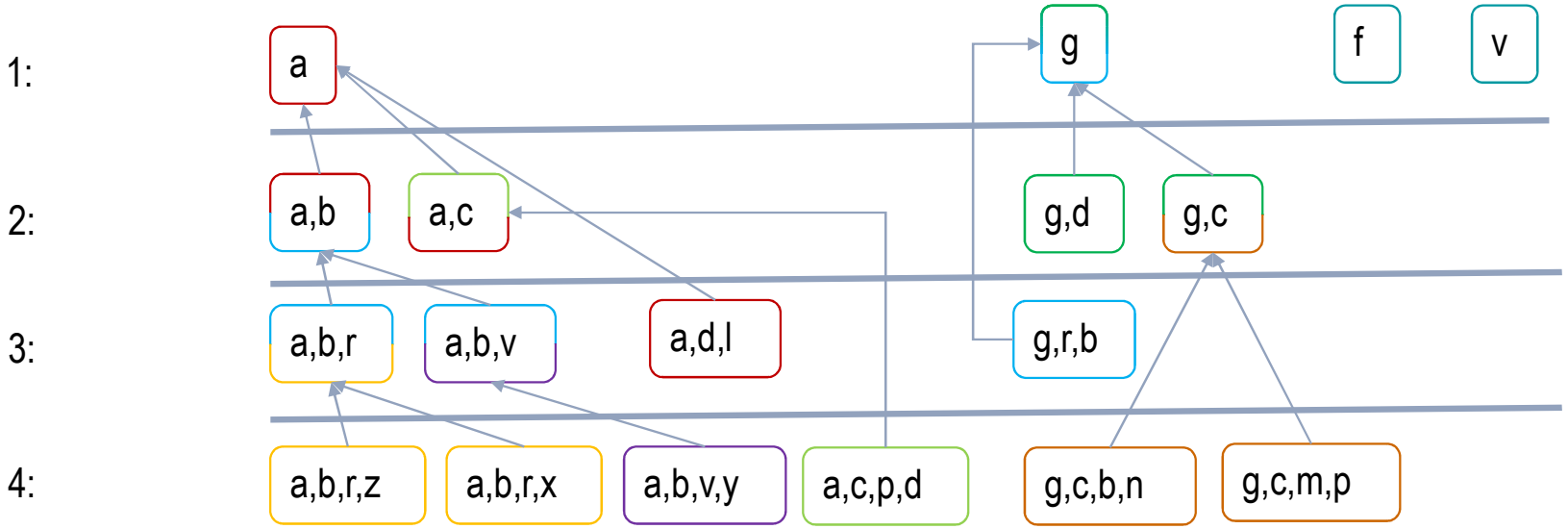
- Laplacian can be replaced, e.g. by symmetric Laplacian
- Alternative: Affinity propagation algorithm

Extracting Representation Patterns from Clusters

What represents a cluster?



Organizing and Combining Patterns



a, a AND (b OR c OR (d AND l))

a AND b AND r, a AND b AND r AND (z OR x)

a AND c, a AND c AND p AND d

a AND b, a AND b AND (r OR v)

a AND b AND v, a AND b AND v AND y

g AND c, g AND c AND
((b AND n) OR (m AND P))

g, g AND (d OR c)

g, g AND r AND b

Cleaning up the results

- Equivalent / already known patterns must be found
- This can be done using an SMT solver (we used z3opt)

```
(declare-const x1 Bool)
(declare-const x2 Bool)
(define-fun  patterneq () Bool
  (= (not (and x1 x2))  (or (not x1) (not x2))))
(assert (not patterneq))
(check-sat)
```

Analysis frequency

- New data arrives steadily → analysis of blocks of new data required
- Analysis frequency is a hyperparameter
- Too often: difficult to find meaningful patterns
- Too rarely: patterns are detected too late or are even suppressed by older ones

Future work

- Extending the toolchain by a (semi-)supervised learning approach
- Maintenance shops get feedback
- Maintenance result can be used as labels for clusters found

Discussion

Thank you for your attention!