Signal Classification using Temporal Logic

Calin Belta
Tegan Family Distinguished Professor
Mechanical Engineering, Systems Engineering,
Electrical and Computer Engineering

Boston University

(with Giuseppe Bombara)

Challenges

- What is bad?

Need (unsupervised) machine learning

- How can we say good from bad?

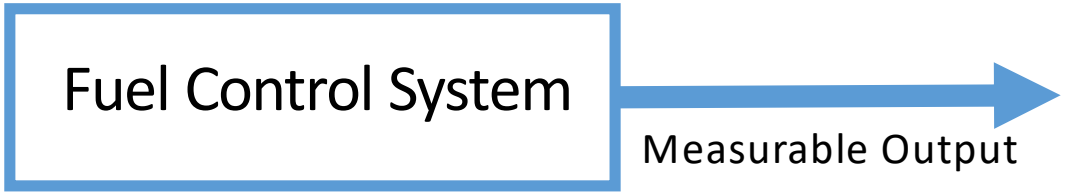
Need quantification of satisfaction and monitoring

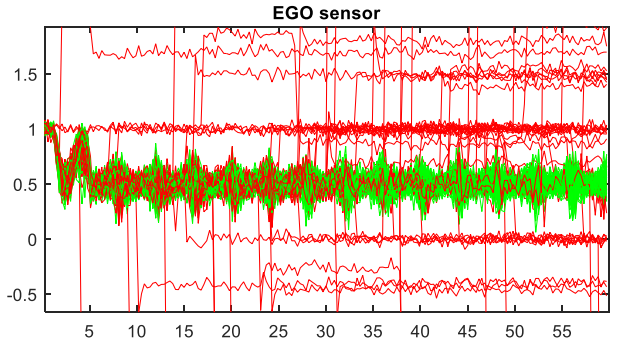
- How can we predict something bad before it actually happens?

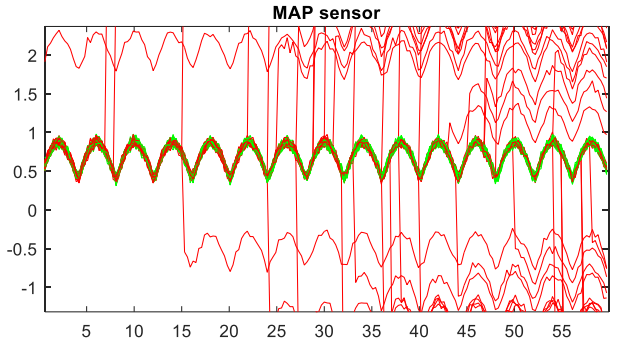
- How can we synthesize control strategy to minimize badness?

Need provably correct optimal control

Problem Formulation







EGO: residual oxygen in the exhaust gas (EGO)
MAP: manifold absolute pressure (MAP)

Given

- a system, e.g., a vehicle component
- data collected during the operation of the system

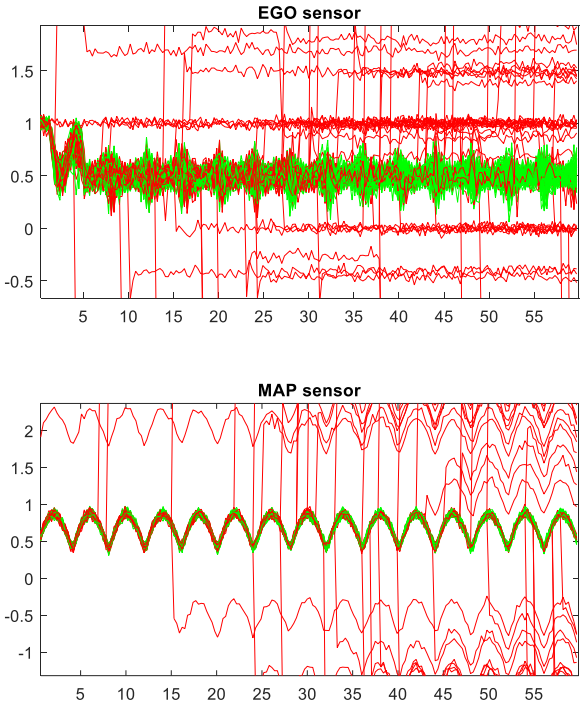
Find

- when something goes bad
- even better, raise a flag before something goes bad
- synthesize a strategy to mitigate badness

No model is available!

Approach

Combined machine learning and formal methods approach: the classifiers are temporal logic formulas



Advantages:

- Easy to interpret classifiers
- The classifiers can be seamlessly combined (composed) with temporal logic specifications to solve synthesis problems
- Our particular approach allows to "quantify" satisfaction and map (supervised, unsupervised, and online) learning to optimization problems
- Potential for monitoring

Classifier for good behavior

$$\mathrm{F}_{[0,60)}((G_{[9.7,59.7)}x_3 < 0.875) \wedge (G_{[0.1,59.7)}x_4 < 0.98) \wedge (G_{[0.5,59.7)}x_4 > 0.29))$$

i.e., "Within 60s, EGO is less than 0.875 for all times in between 9.7s and 59.7s and MAP is less than 0.98 for all times in between 0.1s and 59.7s and MAP is greater than 0.29 for all times in between 0.5s and 59.7s."

Outline

1. Signal Temporal Logic
2. Signal Classification (*supervised learning*)
3. Signal Clustering (*unsupervised learning*)
4. Online Learning
5. Grid TLI
6. Conclusion

Outline

1. **Signal Temporal Logic**
2. Signal Classification (*supervised learning*)
3. Signal Clustering (*unsupervised learning*)
4. Online Learning
5. Grid TLI
6. Conclusion

Signal Temporal Logic (STL)

- STL is a specification language used in the field of formal methods to specify the behavior of hybrid systems (*Maler and Nickovic, 2004*)

- It is a formalism for defining properties over *dense-time real-valued* signals, composed by:
  - The usual Boolean operations $(\text{and } \wedge, \text{or } \vee, \text{not } \neg)$
  - Temporal Operators $(\text{Globally } \mathrm{G}_{[t_0, t_1]}\varphi, \text{Eventually } \mathrm{F}_{[t_0, t_1]}\varphi)$
  - Continuous predicates $(f(s) > c)$

- Examples of STL formulae
  - $\mathbf{G}_{[2,10]}(x < 3.2)$ – x must be less than 3.2 units at all times in the interval $[2,10]$
  - $\mathbf{F}_{[1,15]}\mathbf{G}_{[0,3]}(x > 5)$ – eventually in the time interval $[1,15]$ x must be more than 5 units for 3 seconds

Signal Temporal Logic (STL)

- It supports a *quantitative semantics* (Donze' et al., 2010; Fainekos et al., 2009)
- Parametric version called PSTL (*Asarin et al. 2012*):

PSTL $\psi = \mathbf{G}_{[a,b)}(s_1 \leq c)$, $\theta = [2.5, 0, 1]$ → STL $\phi_\theta = \mathbf{G}_{[0,1)}(s_1 \leq 2.5)$

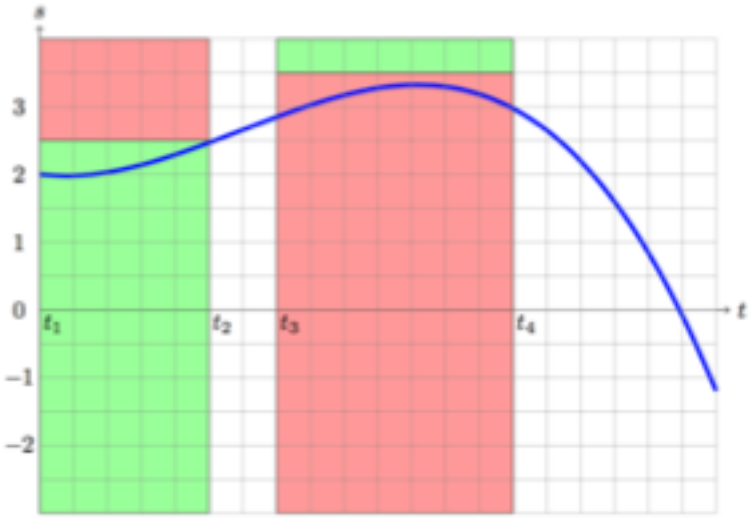
$\mathrm{G}_{[t_1,t_2]}(s \leq 2.5)$ $\quad$ $\mathrm{F}_{[t_3,t_4]}(s > 3.5)$

Boolean: True
Quantitative: 0.01

Boolean: False
Quantitative: -0.2

$\mathrm{G}_{[t_1,t_2]}(s \leq 2.5) \wedge \mathrm{F}_{[t_3,t_4]}(s > 3.5)$

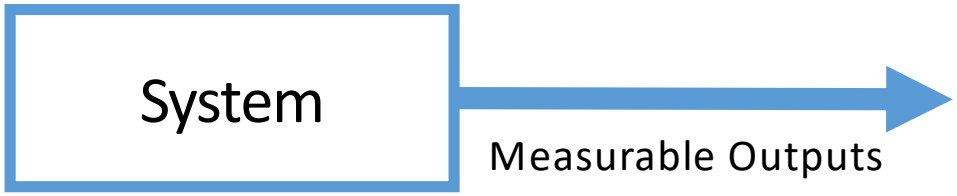
Boolean: False
Quantitative: -0.2

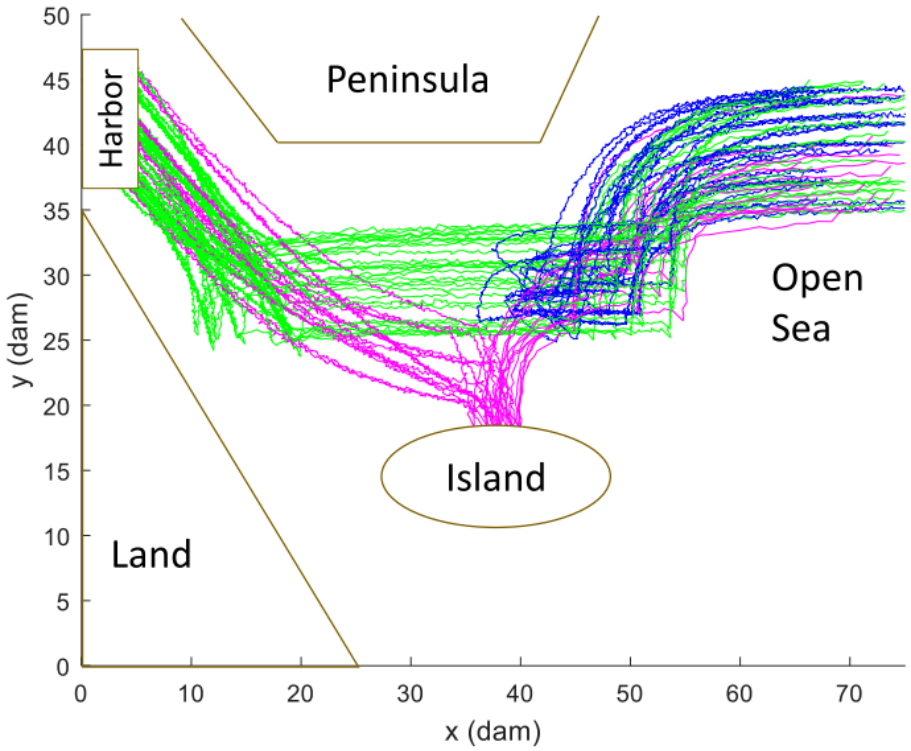


Outline

1. Signal Temporal Logic
2. **Signal Classification (*supervised learning*)**
3. Signal Clustering (*unsupervised learning*)
4. Online Learning
5. Grid TLI
6. Conclusion

Motivating example





Naval surveillance

- Monitor ship traffic and detect anomalies

Given

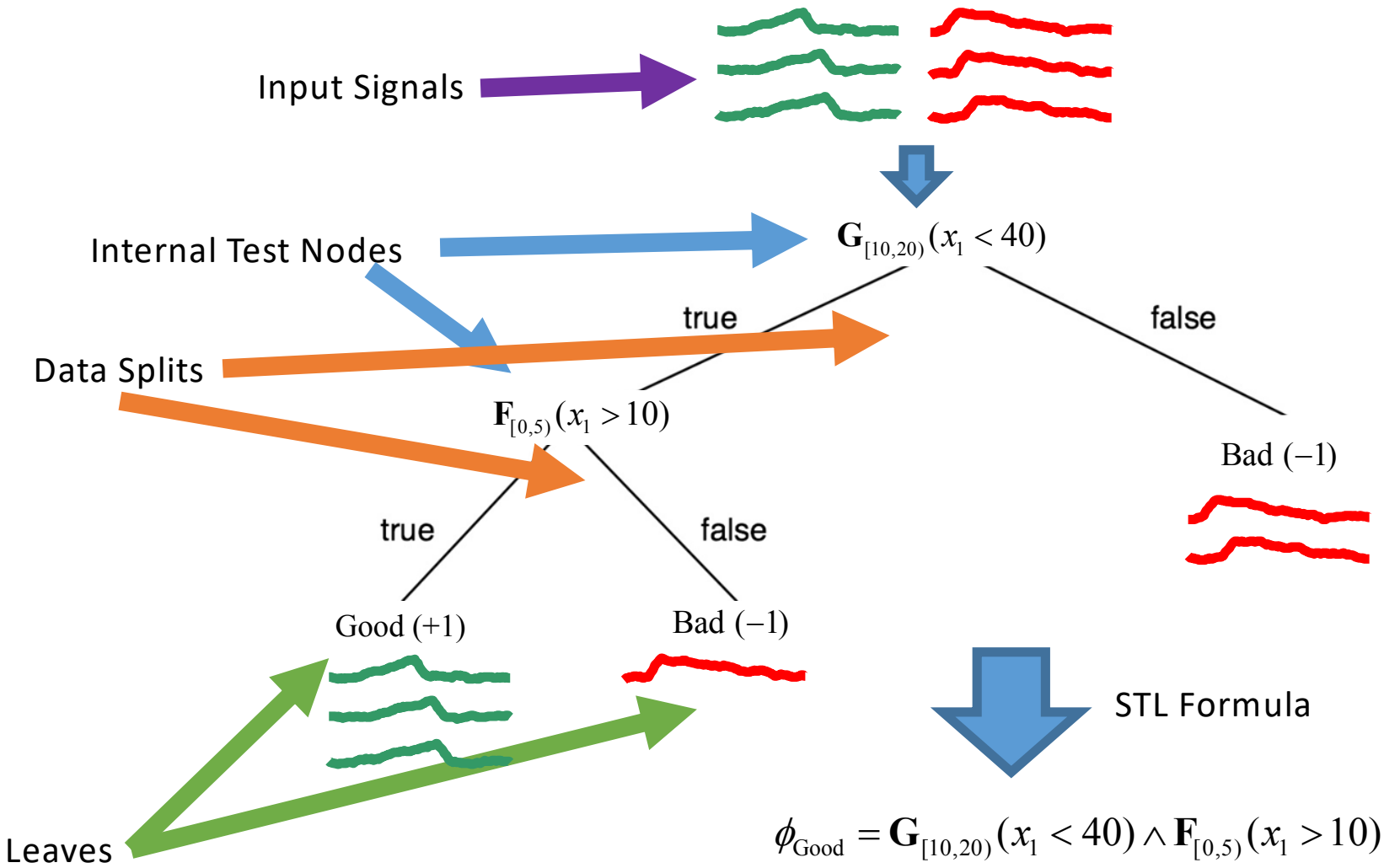
- A system (e.g. a vessel)
- Labelled data collected during its operation (e.g. the vessels trajectories)

The Two-class Classification Problem

- Find an STL formula that distinguishes trajectories produced by a system that exhibit some desired property (e.g., normal vessels) from the other trajectories

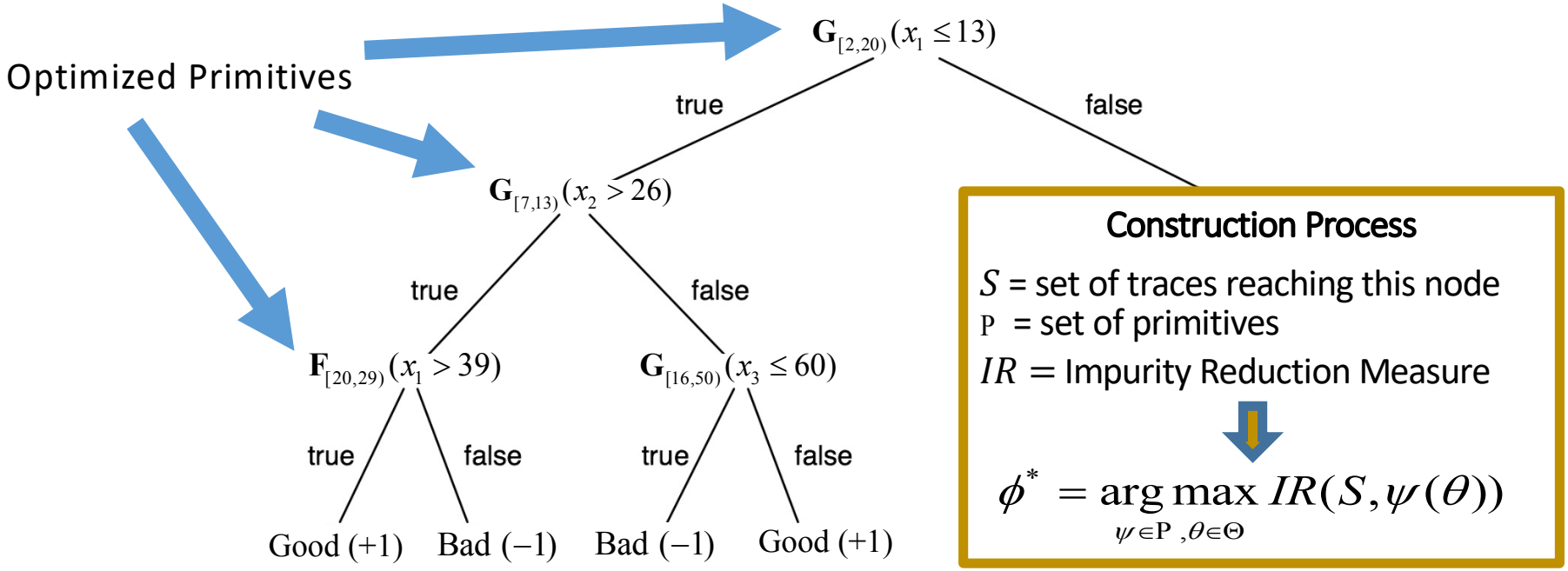
Solution Outline (1)

Key insight: it is possible to build a *map* between a fragment of STL and Special Decision Trees



(Bombara et al., HSCC'16)

Solution Outline (2)



Solution Outline (3)

1. Use simple formulae, called **primitives**, at every node to split the signals

$$\mathcal{P}_1 = \{\mathbf{F}_{[\tau_1,\tau_2)}(x_i \sim \mu) \text{ or } \mathbf{G}_{[\tau_1,\tau_2)}(x_i \sim \mu) \mid i \in \{1,\ldots,n\},\ \sim \in \{\leq, >\}\}$$

$$\mathcal{P}_2 = \{\mathbf{G}_{[\tau_1,\tau_2)}\mathbf{F}_{[0,\tau_3)}(x_i \sim \mu) \text{ or } \mathbf{F}_{[\tau_1,\tau_2)}\mathbf{G}_{[0,\tau_3)}(x_i \sim \mu) \mid i \in \{1,\ldots,n\},\ \sim \in \{\leq, >\}\}$$

2. Rank the primitives using suitably defined (to handle signals) ***impurity reduction measures*** that quantify the quality of the split (Information gain, Gini gain, Misclassification gain, etc.)
   - A good split leads to children that are ***pure*** (contain mostly objects of the ***same class***)

$$H(S) = -\sum_{c \in C} p(S,c,\phi) \log p(S,c,\phi)$$

$$IR(S,\phi) = H(S) - \sum_{\otimes \in \{\text{ï},\perp\}} p_{\otimes} \cdot H(S_{\otimes})$$

Frequency-based weights

$$p_{\top} = \frac{|S_{\top}|}{|S|},\ p_{\perp} = \frac{|S_{\perp}|}{|S|},\ p(S,c;\phi) = \frac{|\{(s^i,l^i) \mid l^i = c\}|}{|S|}$$

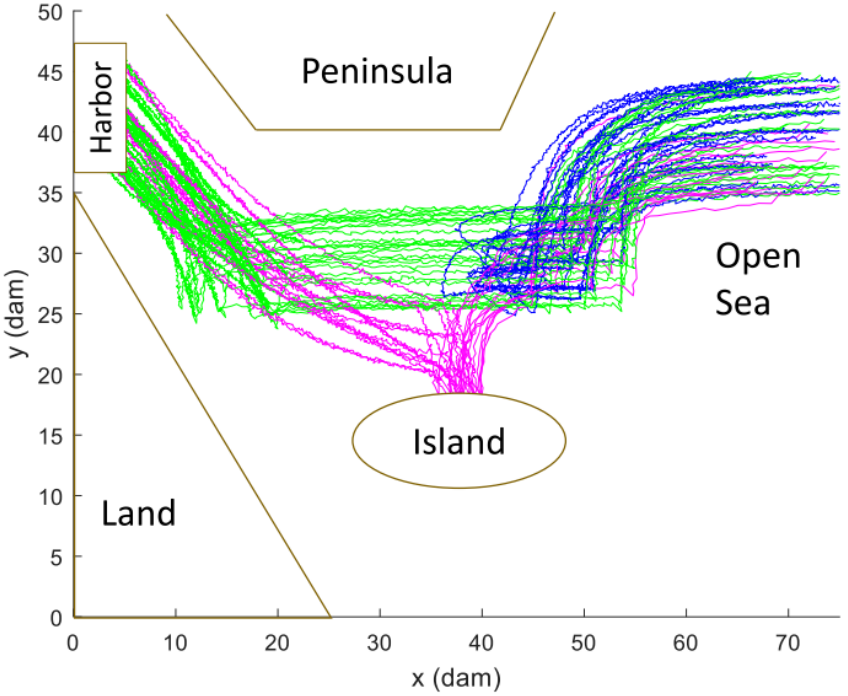
Robustness-based weights (lead to better behaved optimization problems and better classification)

$$p_{\top} = \frac{\sum_{s^i \in S_{\top}} r(s^i,\phi)}{\sum_{s^i \in S} |r(s^i,\phi)|} \qquad p_{\perp} = -\frac{\sum_{s^i \in S_{\perp}} r(s^i,\phi)}{\sum_{s^i \in S} |r(s^i,\phi)|}$$

$$p(S,c;\phi) = \frac{\sum_{s^i \in S_c} |r(s^i,\phi)|}{\sum_{s^i \in S} |r(s^i,\phi)|}$$

Maritime Surveillance – Results

- Framework settings:
  - First-Level primitives P_1
  - Ext. Misclassification Gain MG_r
  - Max Depth *4*, Majority class rate > *0.975*



- Performance Evaluation
  - 5-fold cross-validation
  - Mean Misclassification (MCR) 0.8%
  - Standard Deviation (STD) 0.3%

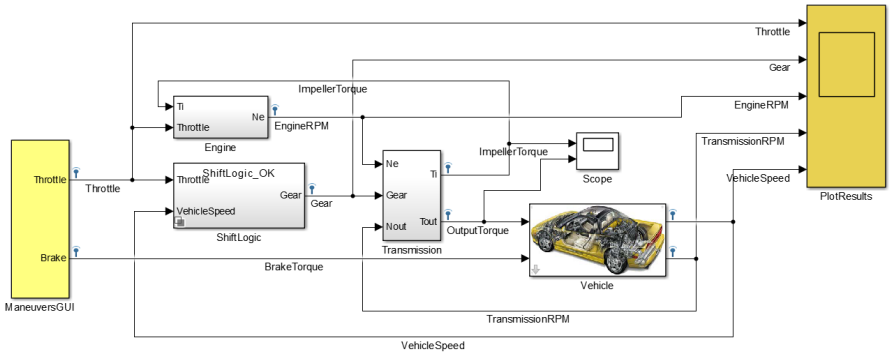
$$\phi_{\text{norm}} = (\mathbf{G}_{[194,202]}x < 19.5 \wedge \mathbf{G}_{[55.2,86]}y > 23.6) \vee$$
$$(\mathbf{F}_{[194,202]}x > 19.5 \wedge ((\mathbf{G}_{[95.6,300]}y < 32.3 \wedge \mathbf{G}_{[161,242]}x < 45.3) \vee$$
$$(\mathbf{F}_{[95.6,300]}y > 32.3 \wedge (\mathbf{G}_{[27.2,299]}y > 30 \wedge \mathbf{G}_{[84.8,220]}x < 56.1))))$$

- A pruning strategy has been implemented to derive shallower trees:
  1. Copes with overfitting on training data
  2. Permits to obtain shorter/simpler formulae

$$\phi_{norm} = (\mathbf{G}_{[194,202]}x < 19.5 \wedge \mathbf{G}_{[55.2,86]}y > 23.2) \vee (\mathbf{F}_{[194,202]}x_1 > 19.5 \wedge \mathbf{G}_{[95.6,300]}x_2 < 32.3)$$

Fuel Control System – Results

- Framework settings:
  - Second-Level primitives
  - Ext. Information Gain
  - Max Depth *3*



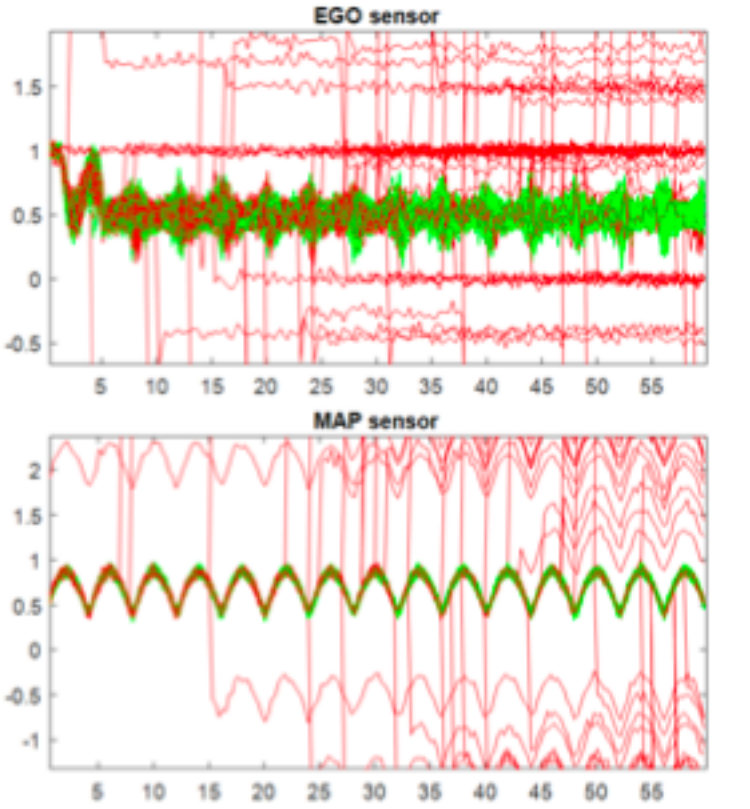
- Performance Evaluation
  - 5-fold cross-validation
  - Mean Misclassification (MCR) 0.054%
  - Standard Deviation (STD) 0.025%

$$\phi^{I_2} = \neg\phi_1^{I_2} \wedge \phi_2^{I_2} \wedge \phi_3^{I_2}$$

$$\phi_1^{I_2} = \mathbf{F}_{[1.85,58.70)}\mathbf{G}_{[0.00,0.57)}(x_1 \leq 0.13)$$

$$\phi_2^{I_2} = \mathbf{G}_{[11.35,59.55)}\mathbf{F}_{[0.00,0.03)}(x_1 \leq 0.99)$$

$$\phi_3^{I_2} = \mathbf{G}_{[1.65,58.89)}\mathbf{F}_{[0.00,0.44)}(x_2 \leq 0.90)$$



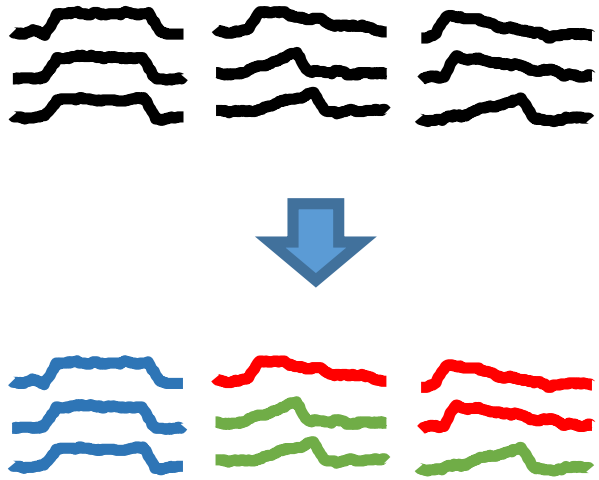
Outline

1. Signal Temporal Logic
2. Signal Classification (*supervised learning*)
3. **Signal Clustering (*unsupervised learning*)**
4. Online Learning
5. Grid TLI
6. Conclusion

The Clustering Problem – Unsupervised learning

- It is the task of *grouping* a set of objects in such a way that objects in the same group (cluster) are more ***similar*** to each other than to the objects belonging to other groups
- No information about the type (or class) of the objects is available in this setting (**labeling is expensive**)
- ***Describe each cluster with an STL formula***

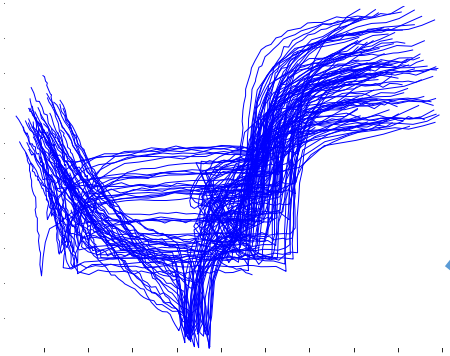
- The interest for the user lies in both
  1. The discriminative power of the groups (*classification*)
  2. The resulting groups themselves (*knowledge discovery*)

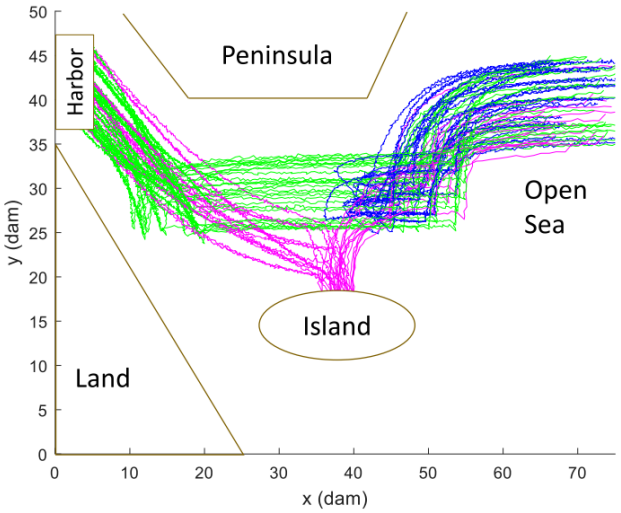


Solutions & Proposed Approach

- **Divisive hierarchical clustering** algorithm that does not require to know the number of clusters beforehand and it does not enforce a probabilistic model to the data
- Construct a binary tree with simple formulae in the internal nodes
- Greedy procedure for construction: split the signals at each node such that the resulting children contain more **similar** signals (***homogeneous***)
- Each leaf is mapped with a cluster and can be translated to a formula

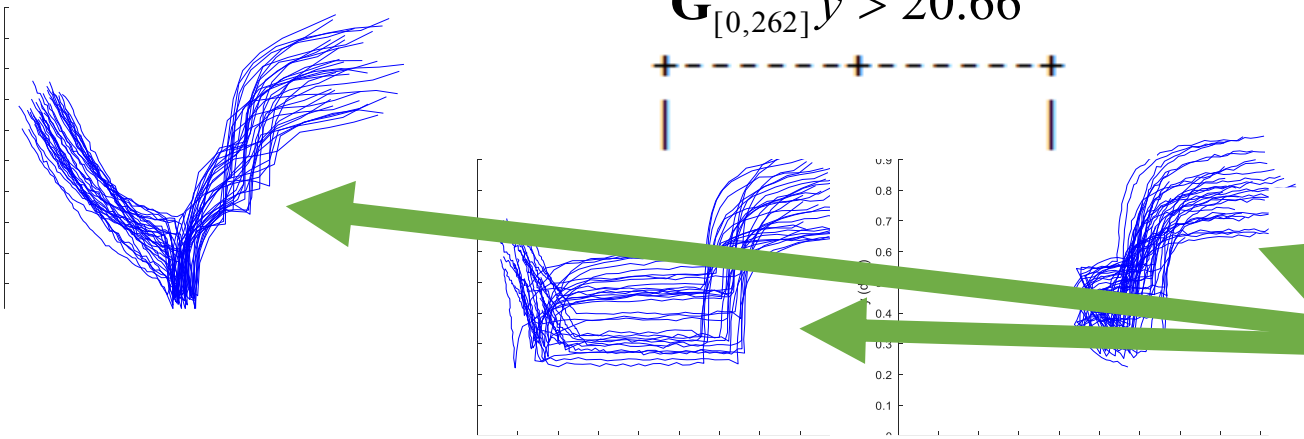
Maritime Surveillance – Hierarchical Clustering





$$\mathbf{G}_{[16.3,280]}x > 24.832$$

$$\mathbf{G}_{[0,262]}y > 20.66$$



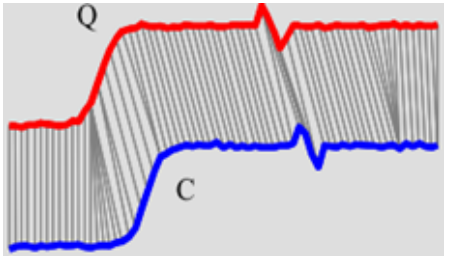
Clusters at the leaves

$$\phi_{norm} = \mathbf{F}_{[16.3,280]}(x \leq 24.832) \wedge \mathbf{G}_{[0,262]}(y > 20.66)$$

MCR<1%

Core Components of the Approach

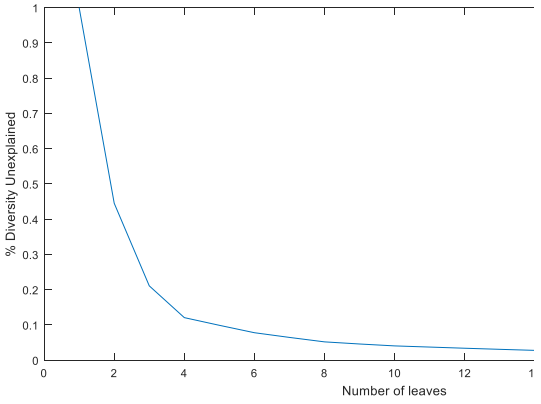
1. A *distance* (or a similarity) between two signals
   - Has to handle the multi-dimensional case
   - Euclidian Distance or Dynamic Time Warping



2. *Inhomogeneity* and *inhomogeneity reduction* measures
   - We want the resulting nodes to be more *homogeneous* (contain more similar signals *within* them) and be different *between* them.
   - Inertia-based homogeneity measure and homogeneity gain:

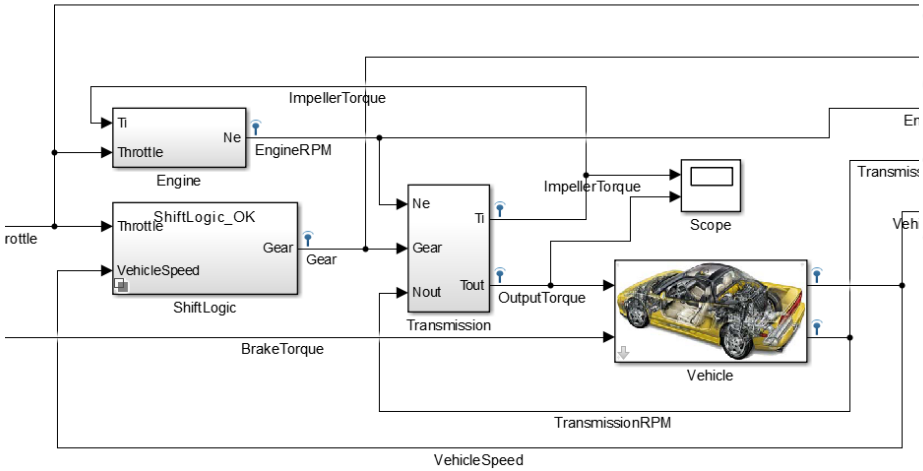
$$I(S) = \frac{1}{2}\frac{1}{|S|^2}\sum_{i=1}^{|S|}\sum_{j=1}^{|S|} d^2(s^i, s^j) \qquad HG(S,\phi) = I(S) - \left[\frac{|S_\top|}{|S|}\cdot I(S_\top) + \frac{|S_\perp|}{|S|}\cdot I(S_\perp)\right]$$

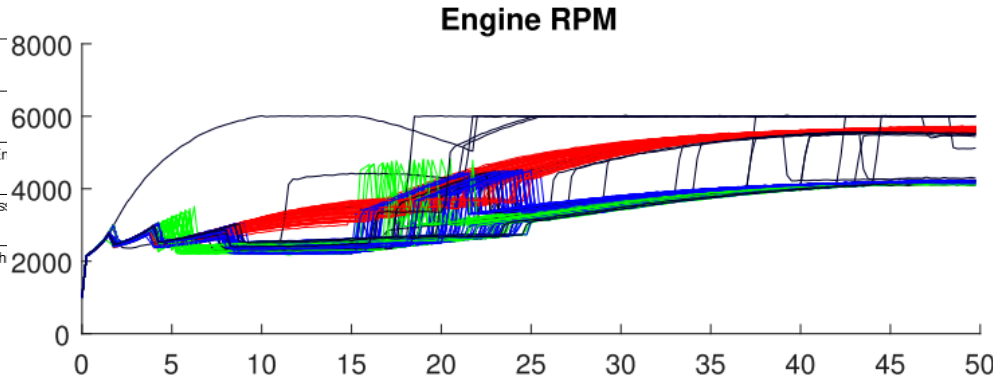
3. A criterion for deciding *where* to grow the tree and *when* to stop
   - Every time a node is split, the *overall unexplained diversity* (inhomogeneity) diminishes
   - **Elbow analysis:** choose the number of clusters so that adding another does not improve much our data model

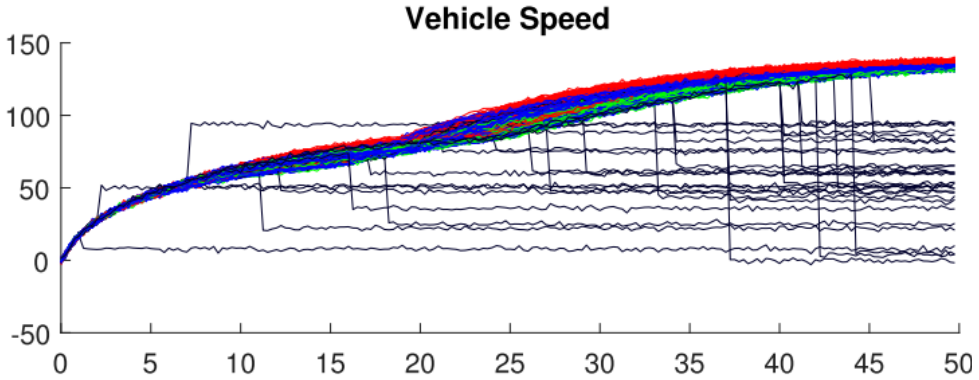


(Bombara and Belta, RV'17)

Automotive Application: Automatic Transmission

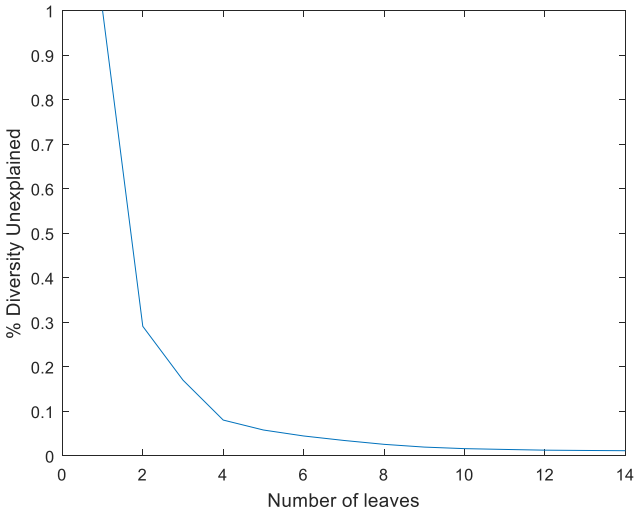




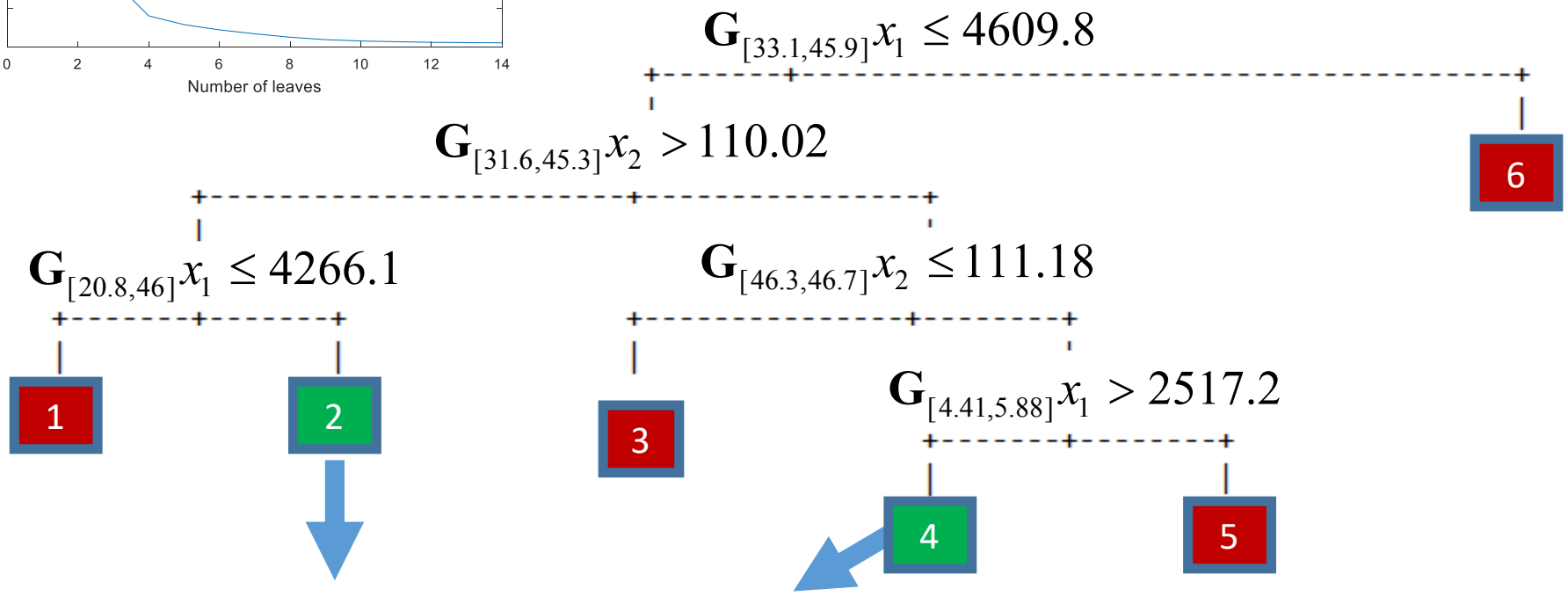


- It is a hybrid system composed of both continuous and discrete state variables
  1. *Engine speed* and *Vehicle speed*
  2. *Gear (1-4)*
- The throttle opening and the brake are the inputs
- A surpass maneuver is simulated and different types of faults were injected
  1. A fault in the vehicle speed sensor (black)
  2. Unable to engage fourth gear (red)
  3. Skip the third gear (green)

Automotive Application – Clustering Results



DTWD Distance
(MaxWarp 20%)



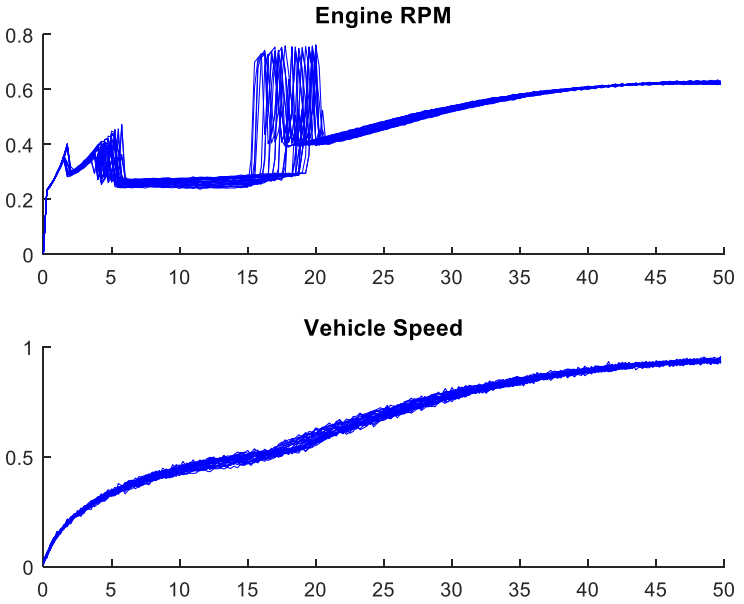
$$\mathbf{G}_{[33.1,45.9]}x_1 \leq 4609.8 \wedge ((\mathbf{G}_{[31.6,45.3]}x_2 > 110.02 \wedge \mathbf{F}_{[20.8,46]}x_1 > 4266.1) \vee$$
$$(\mathbf{F}_{[31.6,45.3]}x_2 < 110.02 \wedge (\mathbf{F}_{[46.3,46.7]}x_2 > 111.18 \wedge \mathbf{G}_{[4.41,5.88]}x_1 > 2517.2)))$$

MCR<3%

Automotive Application – The Clusters

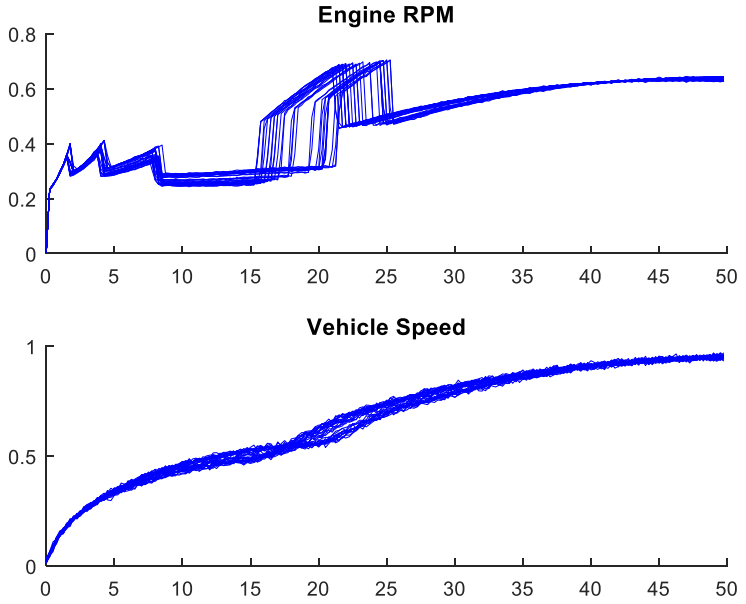
skip the 3rd

1



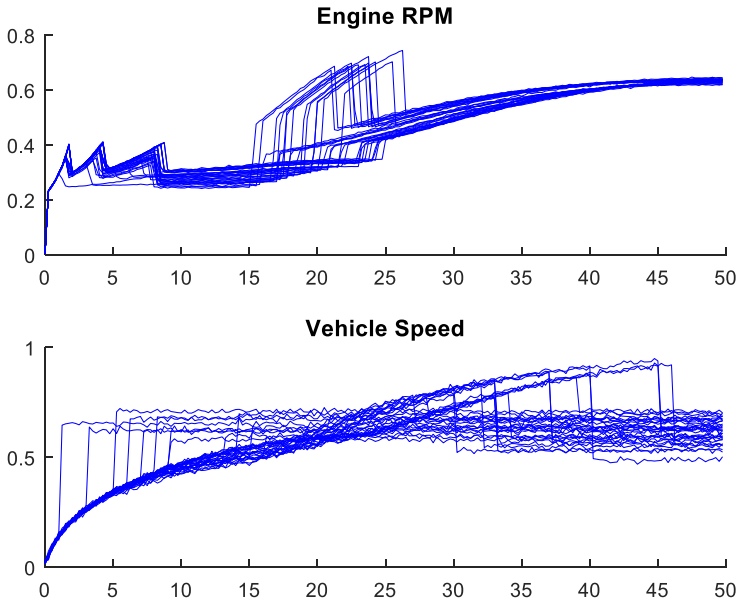
Shifts from 4th to 3rd to 4th

2



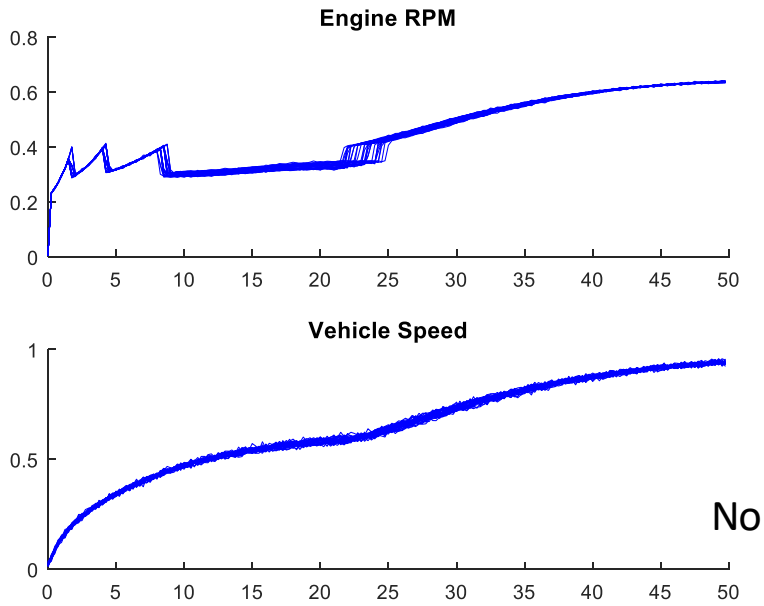
Sensor fault

3



No shifts

4



Outline

1. Signal Temporal Logic
2. Signal Classification (*supervised learning*)
3. Signal Clustering (*unsupervised learning*)
4. **Online Learning**
5. Grid TLI
6. Conclusion

The online learning problem

- It is assumed that new signals become available over time
- The current STL formula should be updated as new data arrives

- **Advantages:**
  - Provides a classifier early during the data collection process and then refines it progressively when more data becomes available
  - Anytime learning: a separation is no longer needed between the *construction phase* and the *deployment phase* of the classifier
- **Challenges:**
  - Finding a way to update a Decision Tree is not trivial, especially if the update has to be done *efficiently*
  - If the arrival of new data causes the change of the best primitive in a node, then that node and all its children have to be pruned and reconstructed from scratch (ITI, *Utgoff et al., 1997*)

A data stream approach

- **Key Insight:** (from data streams) create a new node only when *reasonably* sure about which primitive to use (*Domingos and Hulten, 2006*)
- **Remember**: the best primitive to choose at each node is the one with the highest impurity reduction (IR) value
- **Thought experiment:** if infinite data S_∞ was available, and ψ_1 and ψ_2 were the best and the second best primitives respectively, we would have

$$\Delta IR(S_\infty) = IR(S_\infty, \psi_1) - IR(S_\infty, \psi_2) > 0$$

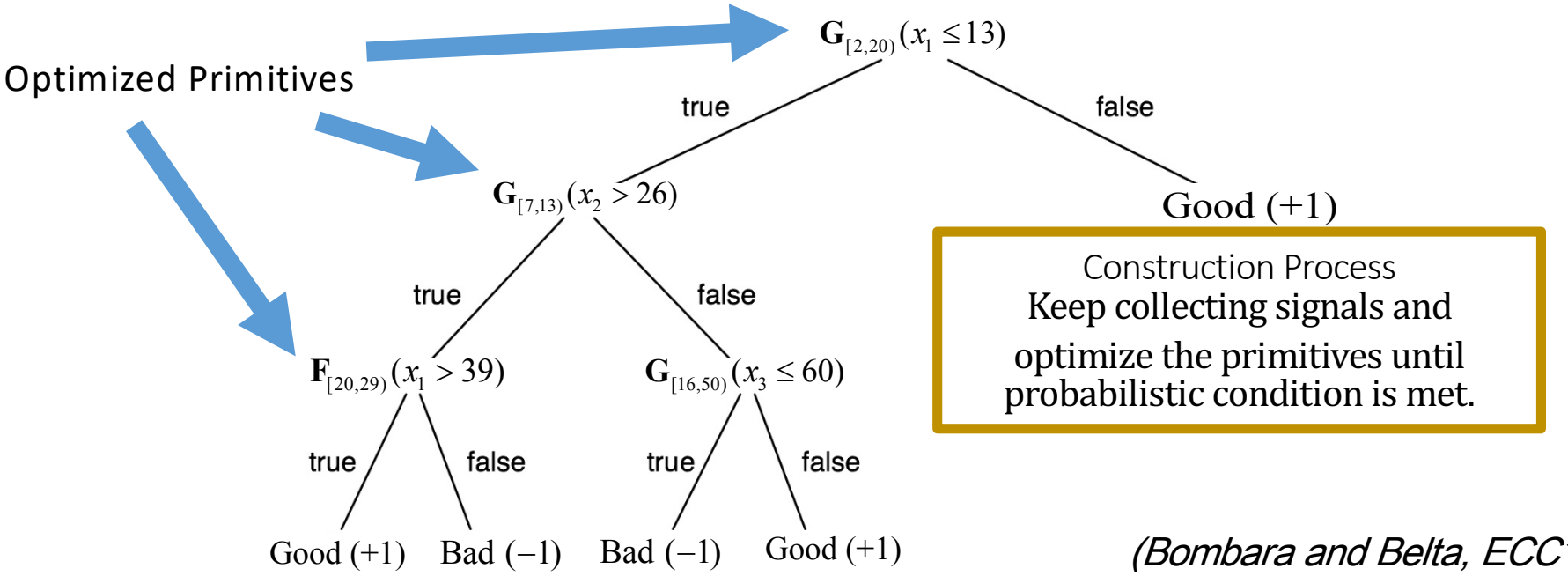
- With a finite amount of data S_N, we cannot be sure about the best formula. However a **probabilistic assessment** on the best primitive can be made:

$$\text{If} \quad \Delta IR(S_N) > \varepsilon(N, \delta) \quad \text{then} \quad \Pr(\Delta IR(S_\infty) > 0) \geq 1 - \delta$$

- If the difference in IR on finite data is greater than a certain threshold ε, then with probability $1 - \delta$ the primitive ψ_1 is indeed the best choice
- Probabilistic inequalities and approximation techniques have been exploited over time to devise a value for the threshold ε (Hoeffding, McDiarmid, etc) (*Rutkowski et al., 2015*)

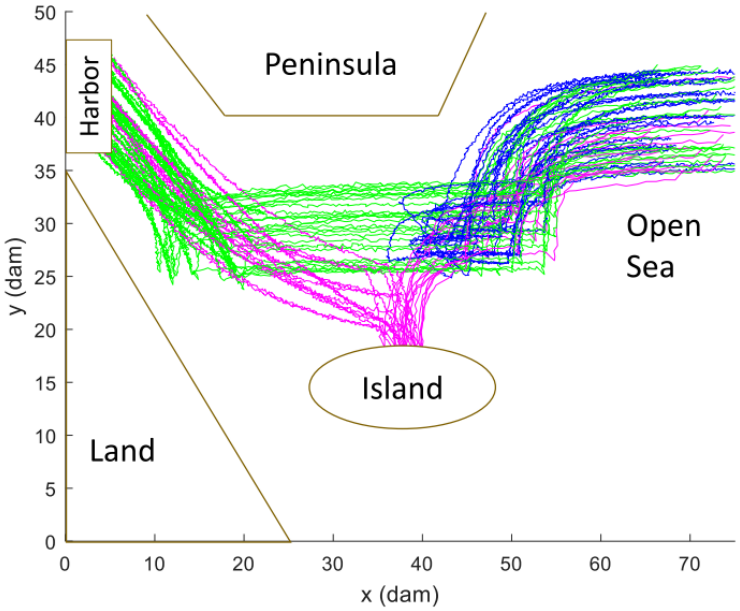
Outline of the online learning algorithm

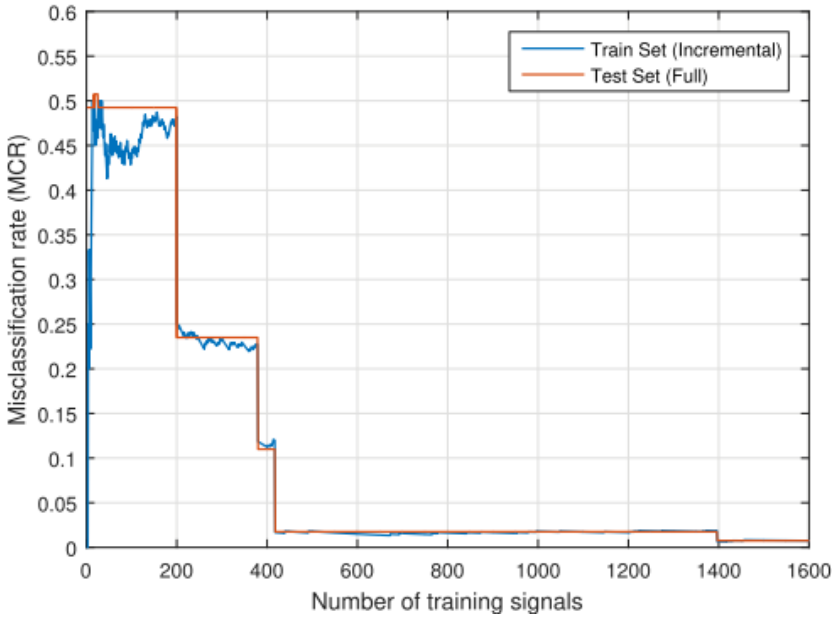
1. Receive a new signal s to process
2. Sort the new signal s through the tree to the leaf L where it belongs
3. Update the statistics and optimize the candidate primitives of leaf L
4. Compute the impurity reduction IR for all primitives at leaf L
5. If $\Delta IR(S_N) > \epsilon(N, \delta)$ is satisfied, split the leaf L using the best primitive



(Bombara and Belta, ECC'18)

Maritime Surveillance





- Using a 5-fold CV, we obtained a mean MCR of 1.45% (STD .88%)
- A sample formula obtained after processing 1600 signals is:

$$\phi = (\phi_1 \wedge \phi_2) \vee (\neg\phi_1 \wedge (\phi_3 \vee (\neg\phi_3 \wedge \phi_4)))$$
$$\phi_1 = \mathbf{G}_{[211,295)}(x \leq 18.6) \qquad \phi_2 = \mathbf{G}_{[69.7,158)}(y > 23.7)$$
$$\phi_3 = \mathbf{G}_{[97.9,299)}(y \leq 32.4) \qquad \phi_4 = \mathbf{F}_{[61,178)}(x \leq 22.2)$$

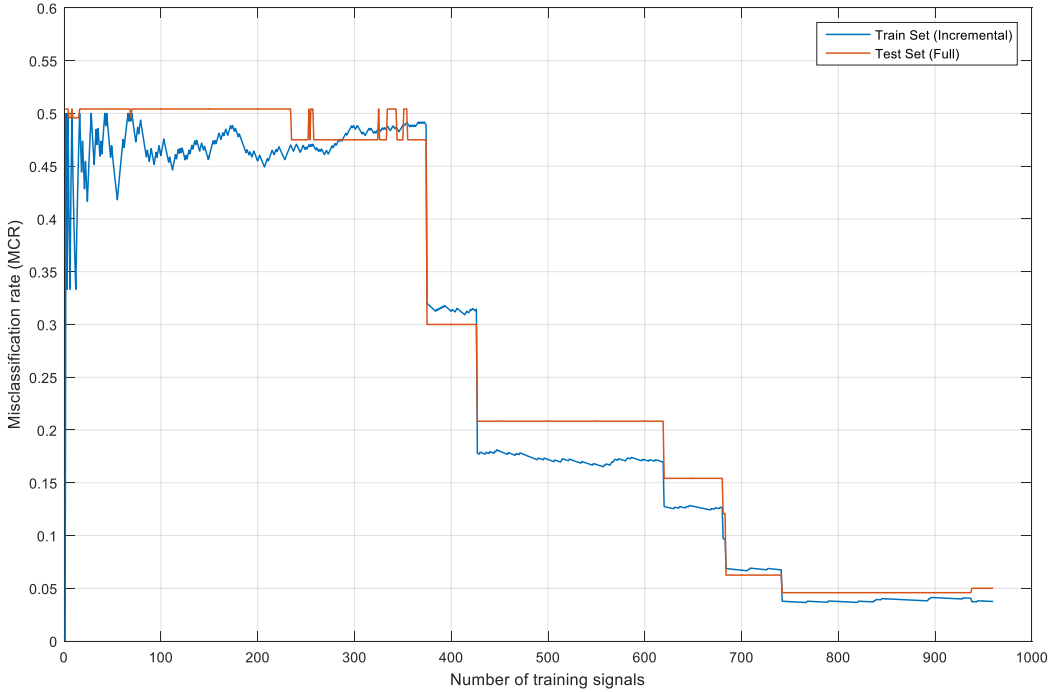
- The evolution of the classification accuracy indicates if a satisfactory solution has already been obtained and there is little reason to continue
- If we stop after 420 signals, we obtain the following (simpler) formula:

$$\phi = (\phi_1 \wedge \phi_2) \vee (\neg\phi_1 \wedge \phi_3)$$

with an associated misclassification rate of 2.5%.

Fuel Control System

- The goal of this system is to maintain the air-to-fuel ratio close to the ideal value in the combustion process (Simulink Built-in model)
- One output, the air-to-fuel ratio, one control variable, the fuel rate, two inputs, the engine speed and the throttle command, and two main sensors, the residual oxygen in the exhaust gas (EGO) and the manifold absolute pressure (MAP)
- Random Gaussian noise has been added at various points in the system.
- Faults have been injected with a **random** arrival time and a **random** offset



- MCR of 2.17% (STD 1.16%)
- After a certain point, using more signals increases the complexity of the formula but does not improve much its classification accuracy
- If the MCR for the test set increases, it is a sign of **overfitting**

Outline

1. Signal Temporal Logic
2. Signal Classification (*supervised learning*)
3. Signal Clustering (*unsupervised learning*)
4. Online Learning
5. **Grid TLI**
6. Conclusion

Grid TLI

Find an STL formula such that all the signals in a set S satisfy the formula.

- the STL formula should fit the signals,
- The fit should be neither too tight nor too loose
- A loose fit results in an STL formula that is not very useful, since it might not

grasp the peculiar characteristics of the signals.

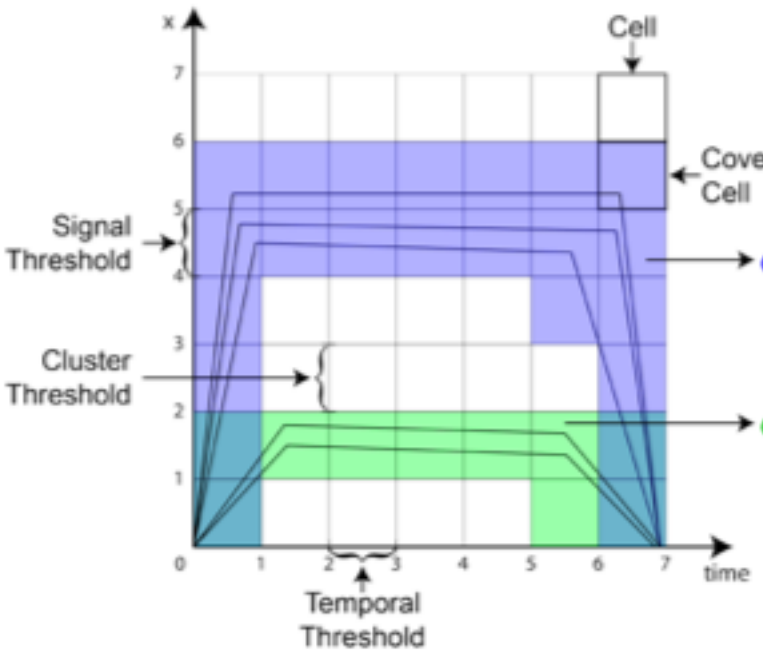
- In contrast, a tight fit results in a long STL formula that is excessively

specific to the training signals.

- Ideally, the granularity ofthe fit of the STL formula must be tunable based on the requirements and properties of the system.

Vaidyanathan et.al., Grid-Based Temporal Logic Inference, CDC 2017

Grid TLI



$$\phi_1 = \mathbf{G}_{[0,7]}(x \leq 6) \wedge \mathbf{G}_{[0,1]}(x > 0) \wedge \mathbf{G}_{[1,5]}(x > 4) \wedge \mathbf{G}_{[5,6]}(x > 3) \wedge \mathbf{G}_{[6,7]}(x > 0)$$

$$\phi_2 = \mathbf{G}_{[0,7]}(x \leq 2) \wedge \mathbf{G}_{[0,1]}(x > 0) \wedge \mathbf{G}_{[1,5]}(x > 1) \wedge \mathbf{G}_{[5,7]}(x > 0)$$

$$\phi = \phi_1 \vee \phi_2$$

1) the signal space is partitioned with a grid;

2) the signals are distributed into separate clusters;

3) Simple STL primitives are fit to each cluster; and

4) the clusters are then mapped to an STL formula.

Algorithm 1: GRID-TLI

input : S, x_t, t_t, c_t

output: ϕ

1: Γ = ConstructGrid(S, x_t, t_t)

2: $(S_1, \ldots, S_k)$:= ClusterSignals(S, c_t)

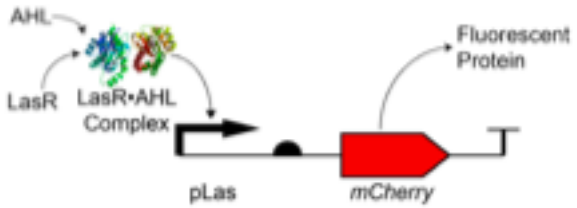
3: Γ_i := FindCoveredCells(Γ, S_i), $i = 1, \ldots, k$

4: ϕ_i := ClusterToFormula(Γ_i), $i = 1, \ldots, k$

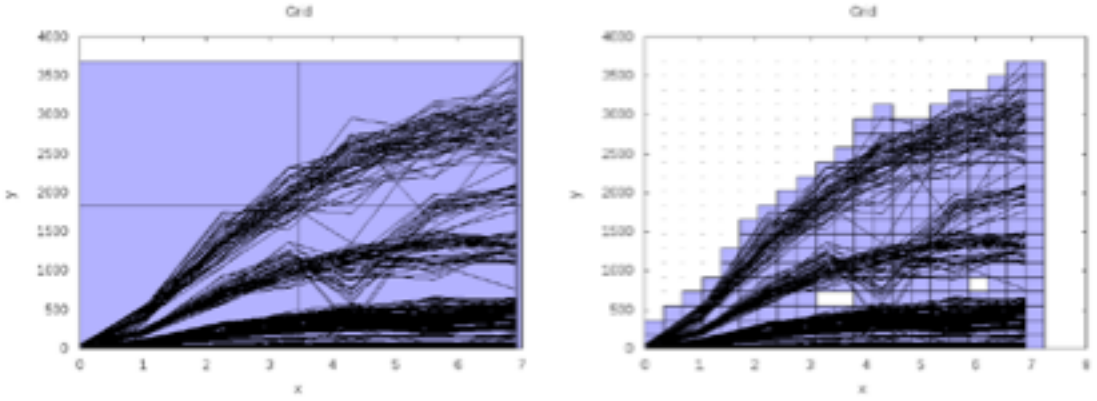
$\phi := \bigvee_{i=1}^{k} \phi_i$

Vaidyanathan et.al., Grid-Based Temporal Logic Inference, CDC 2017

Grid TLI

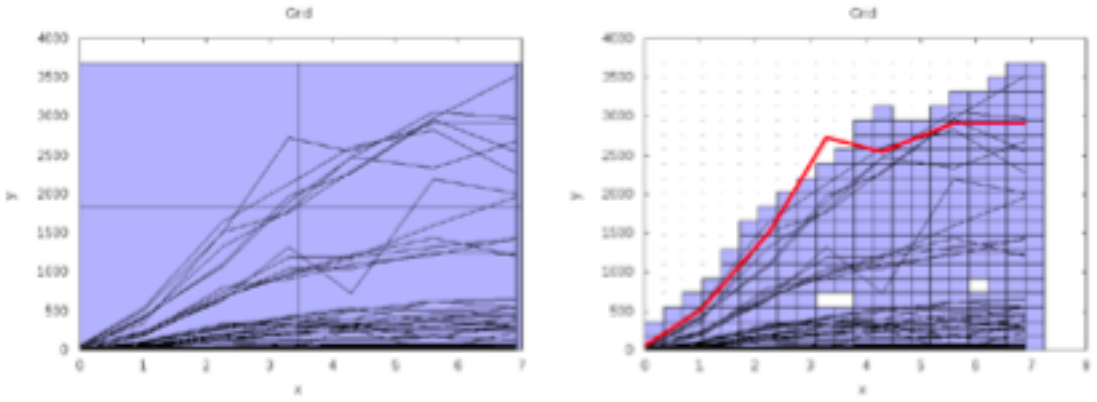


$$\mathbf{G}_{[0,6.9]}(x \leq 3687) \wedge \mathbf{G}_{[0,6.9]}(x \geq 0).$$ Formula too large to show



(a) Large Thresholds - Training

(b) Small Thresholds - Training



(c) Large Thresholds - Testing

(d) Small Thresholds - Testing

$(x_t = 1843.5,\ t_t = 3.45,\ c_t = 1843.5),$ $(x_t = 184.35,\ t_t = 0.345,\ c_t = 0).$

Vaidyanathan et.al., Grid-Based Temporal Logic Inference, CDC 2017

Outline

1. Signal Temporal Logic
2. Signal Classification (*supervised learning*)
3. Signal Clustering (*unsupervised learning*)
4. Online Learning
5. Grid TLI
6. **Conclusion**

Conclusion

- We introduced an inference framework of temporal logic properties from data
- It provides customizable decision-tree algorithm***s*** that output STL formulae as data classifiers
- We developed *supervised*, *unsupervised*, *offline,* and *online* variants of the learning procedures
- We have applied these algorithms to solve problems in maritime surveillance and automotive fields

Ongoing work

- Perform a comprehensive comparative study of the framework for multiple choices of the set of primitives and the impurity measures
- Improve the local optimization
  - It will boost the overall performance of the algorithms
  - Exploit the *monotonicity* property of some PSTL primitives
- Learning with online monitoring