Calculation of Decision Tree Similarity/Distance

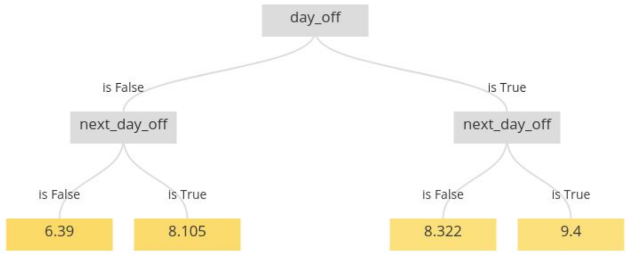
Fengli LIN

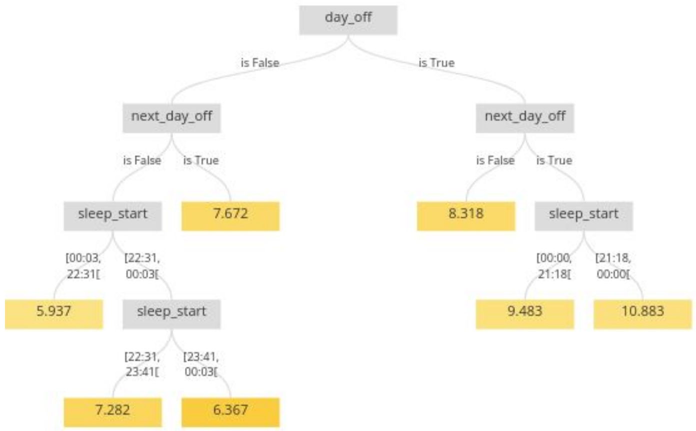
Introduction

Definition:

Given two Decision Trees, we want to calculate a value to represent the degree of similarity between them.

(To be notice that, the value here is mainly used to demonstrate tree A is more similar to tree B than tree C. Because we don't have a golden label for the value itself.)





Why we want to calculate the distance?

Usage:

- Track the evolving process of agents (measuring model/data stability, debugging)
- Detect anomaly(in the case of sudden change of data)
- Clustering agents(regroupe users)
- Transfer existing models to new agents/users which have little data

Methodology

1. Structural similarity
    - Edit distance method
    - Substructure method
2. Semantic similarity
    - Predict value method

Structure: Edit distance

Definition:

The tree edit distance between ordered labeled trees is the **minimal-cost sequence of node edit operations** that transforms one tree into another. We consider following three edit operations on labeled ordered trees:

- **delete** a node and connect its children to its parent maintaining the order.
- **insert** a node between an existing node and a subsequence of consecutive children of this node.
- **rename** the label of a node.

There are many different sequences that transform one tree into another. We assign a cost to each edit operation. Then, the cost of an edit sequence is the sum of the costs of its edit operations. Tree edit distance is the sequence with the minimal cost.

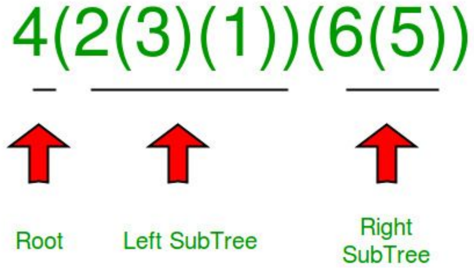
Structure: Edit distance

Algorithm:

Algorithm 1 Syntactic Similarity algorithm (SySM) [38]

INPUT: Group T of n induced decision trees, where n = number of dataset slices
OUTPUT: Chosen Decision Tree T_{cm} where $cm \in [1..n]$

if T is null **then**
  return *failure*
end if
for all $i = 1$ to n **do**
  Set NM = list of node attribute names of $\{Ti\}$ (the nodes are in DFS order)
  Set $TBF = \{\}$, TBF is Tree Bracket Format
  Call UpdateNodes($NM(1)$), $NM(1)$ is root of the tree
  Set $TBF_i = TBF$
end for
for all i=1 to n **do**
  for all j=1 to n **do**
    Set $DM_{ij} = RTED\ (T_i, T_j)$ ▷ // DM_{ij} is Distance Matrix, $RTED$ is Robust Tree Edit Distance
  end for
end for



Structure: Edit distance

Problem:

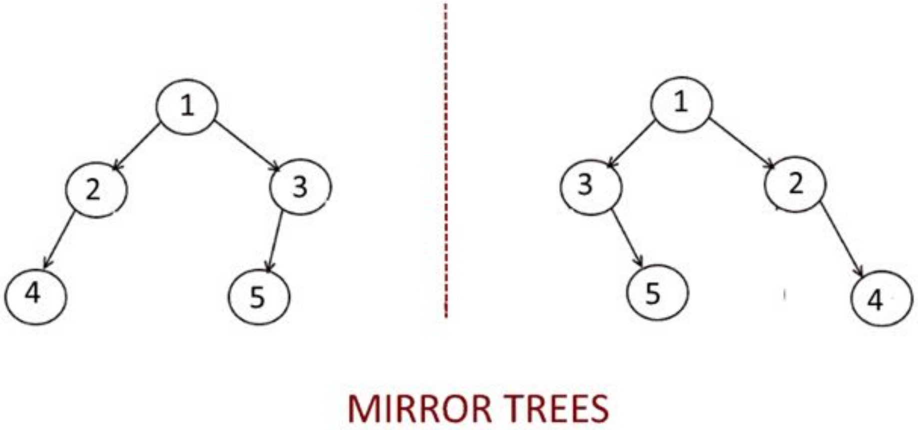
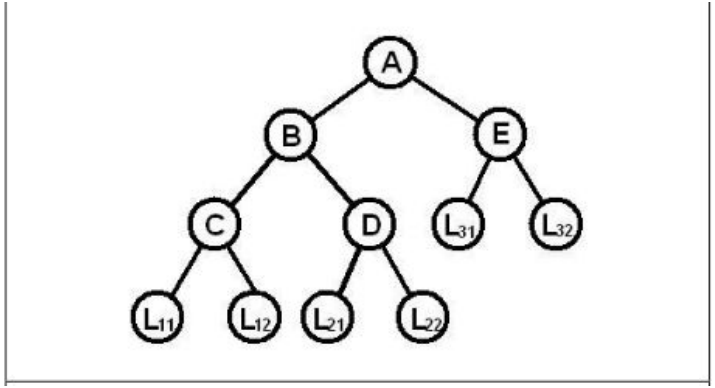


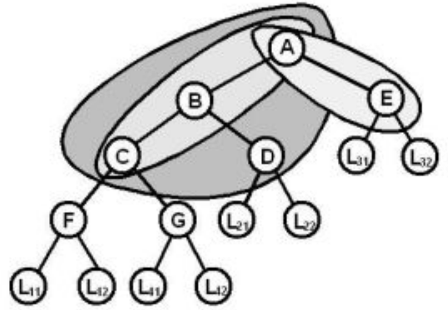
Figure 1 Mirror trees

Structure: Substructure

Idea:

Focus on the similarity of substructures between two decision trees.





Structure: Substructure

Algorithm:

1. Linearize the paths of DTs into sequences of branches(BSs). The tree sequences(TSs) consist of a set of BSs.
2. Cluster those branch sequences by their output labels
3. Apply a sequential pattern mining algorithm(prefix-span algo) to find all frequent sequences(FSs)
4. Calculation of Similarity

$$Sim(FS_i, FS_j) = 100 \times \frac{\sum_{x=1}^{m} \frac{\#(FS_{ic_x} \cap FS_{jc_x})}{Max(|FS_{ic_x}|, |FS_{jc_x}|)}}{m}$$

Structure: Substructure

Example:

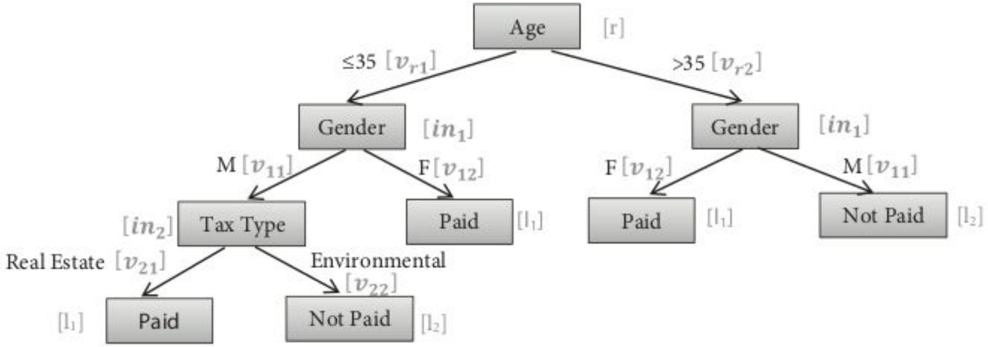


Figure 2. An example tree structure related to municipal data used in this study.

$$BS_1 = \langle([Age] = [\leq 35])\,([Gender] = [M])\,([Tax\,Type] = [Real\;Estate])\,([Paid])\rangle$$

$$BS_2 = \langle([Age] = [\leq 35])\,([Gender] = [M])\,([Tax\,Type] = [Envrn.])\,([Not\;Paid])\rangle$$

$$BS_3 = \langle([Age] = [\leq 35])\,([Gender] = [F])\,([Paid])\rangle$$

$$BS_4 = \langle([Age] = [>35])\,([Gender] = [F])\,([Paid])\rangle$$

$$BS_5 = \langle([Age] = [>35])\,([Gender] = [M])\,([Not\;Paid])\rangle$$

$$Frequent\;Sequence\;for\;TS_1 = \langle\{([Gender] = [M])\,([Tax\,Type] = [Real\;Estate])\rangle$$

$$Frequent\;Sequence\;for\;TS_2 = \langle\{([Gender] = [M])\,([Tax\,Type] = [Real\;Estate])\rangle\}$$

Semantic: Predict value

Idea:

Only focus on the predict value, regardless of the detailed structure. Because DTs, though may be structurally different, may describe the same concept, i.e., they may be semantically similar or even identical to each other

Semantic: Predict value

Algorithm:

The agreement of two classifiers is defined as the probability of producing the same prediction over instances drawn from U (A).

```
# statistic calculation
res1 = np.asarray(result1)
res2 = np.asarray(result2)
diff_arr = np.abs(res1-res2)/res1
diff_mean = np.mean(diff_arr)
diff_std = np.std(diff_arr)
similarity = 1 - diff_mean - diff_std
```

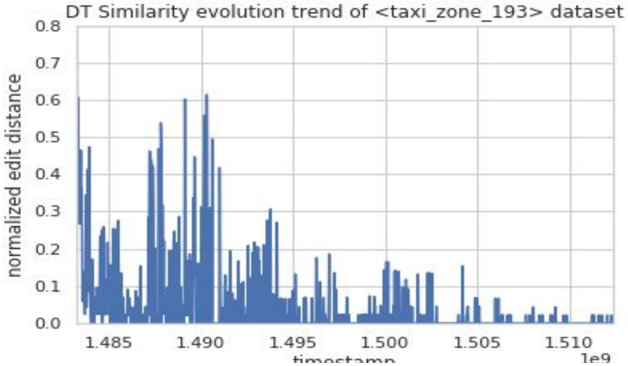
Semantic: Predict value

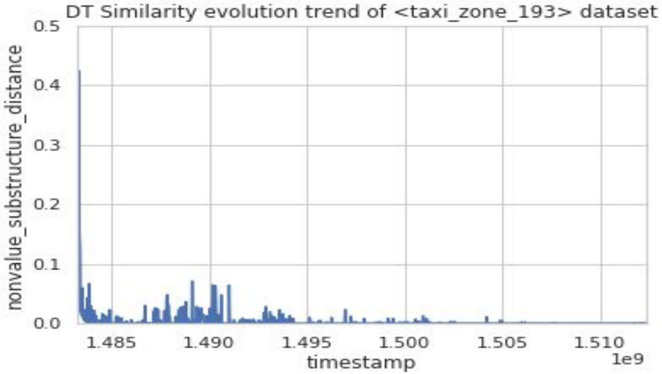
Problem:

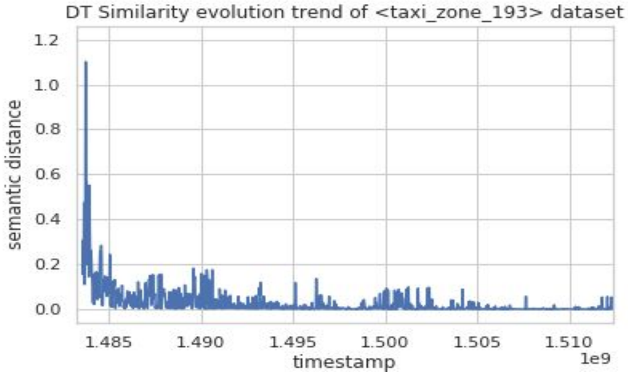
The choice/quality of test dataset

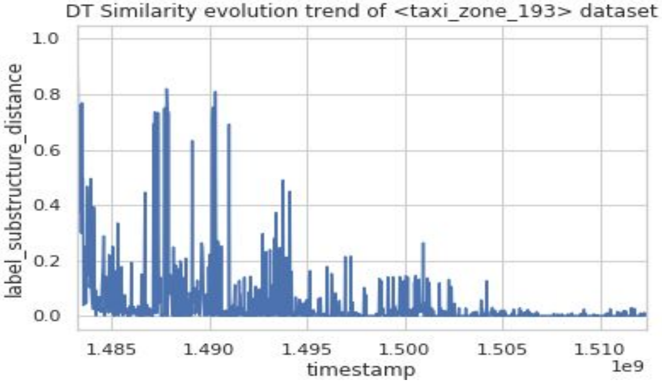
Experiments: Evolving process

~~NYC Tax~~i and Limousine Commission (LTC) dataset









Experiments: Evolving process

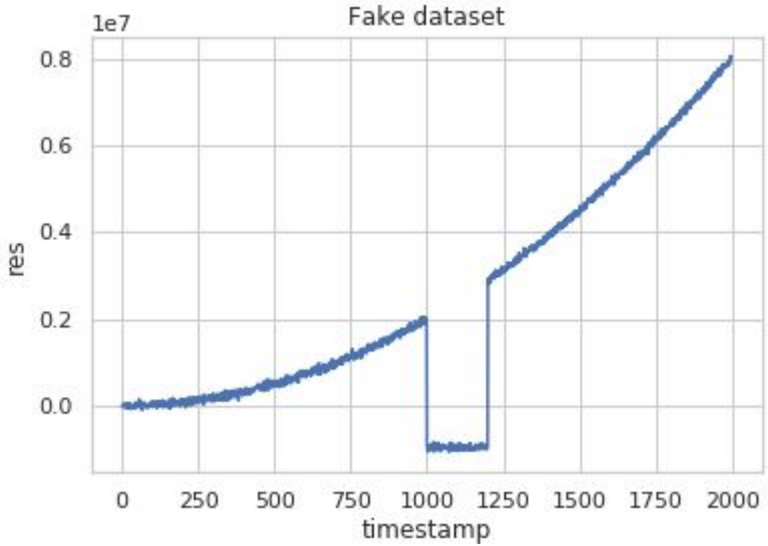
NYC Taxi and Limousine Commission (LTC) dataset

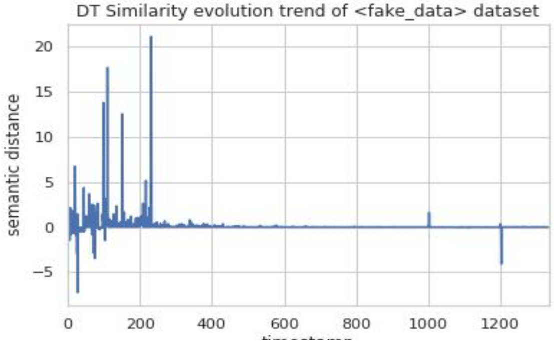
Methods	Edit distance	Nonvalue_Substructure	Nonconti_Substructure	Label_Substructure	Predicted values
Time used	52.785390948	9.6203202	31.281042096	41.140227564999996	578.519689588

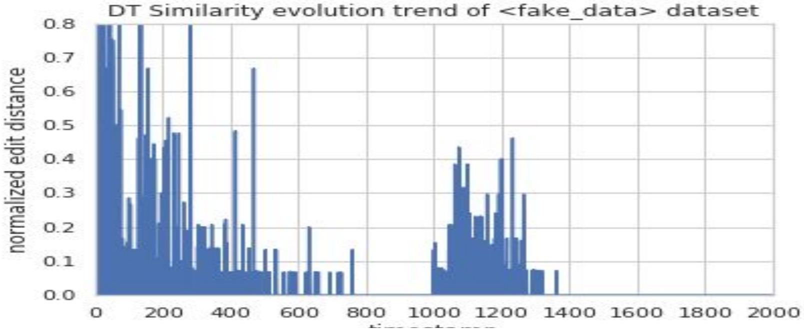
Experiments: Detect anomaly

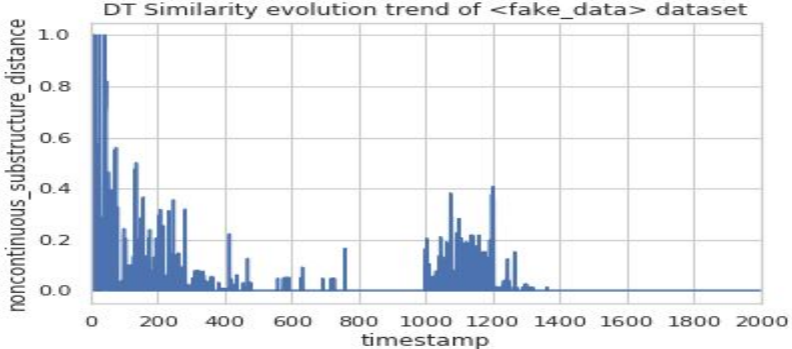
Fake dataset

(change of data distribution between [1000,1200])





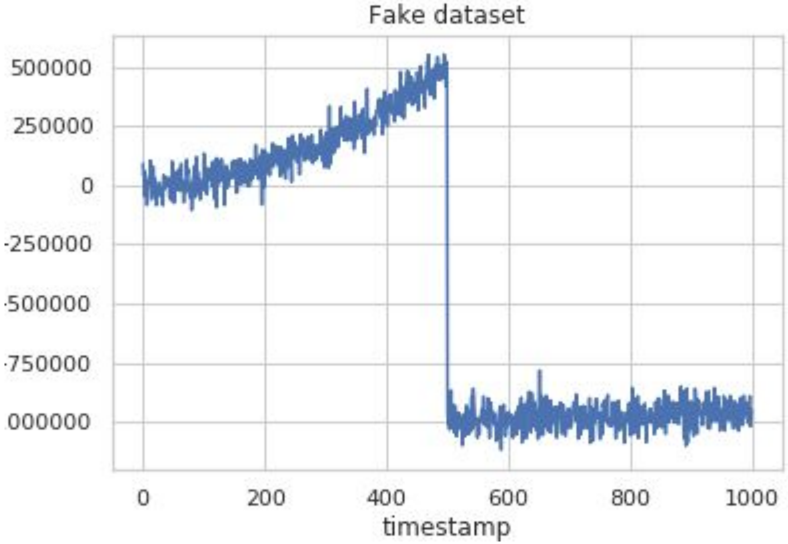


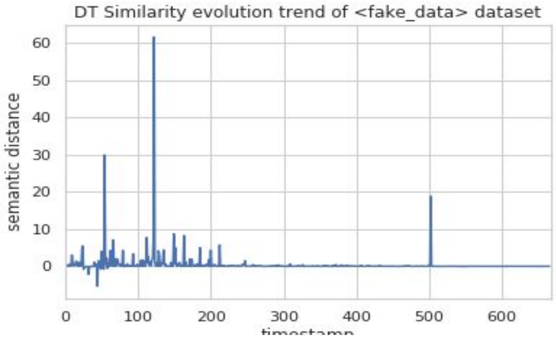


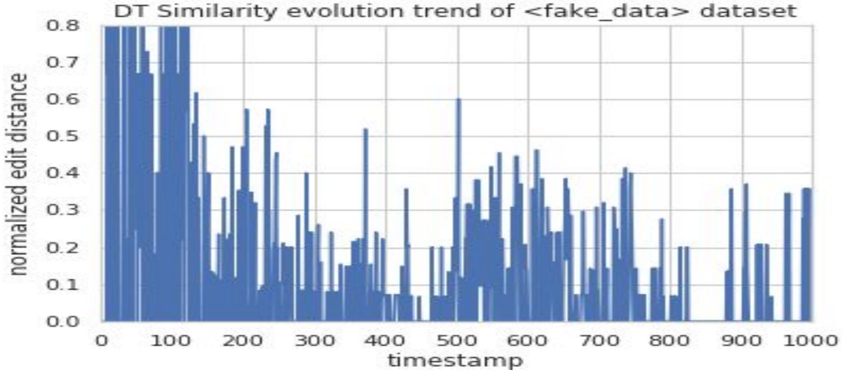
Experiments: Detect anomaly

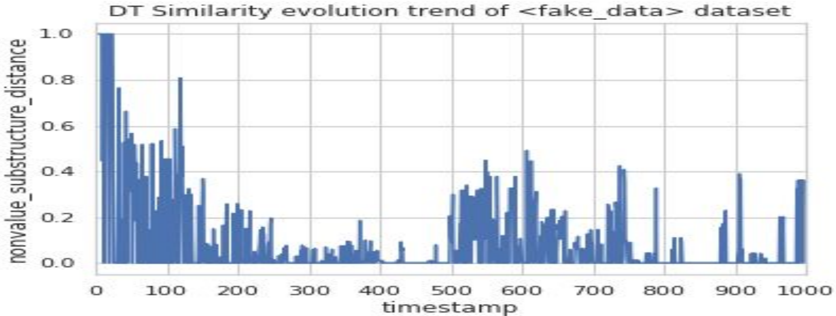
Fake dataset

(change of data distribution at 500)







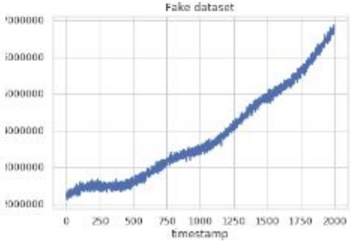


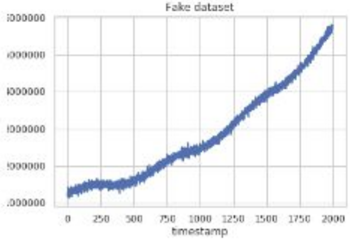
Experiments: Clustering

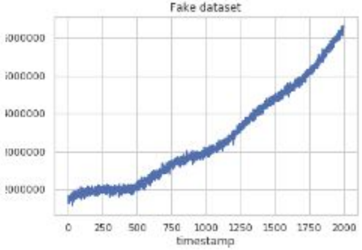
Fake dataset

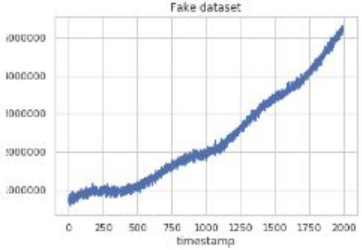
(similar distributi on but different

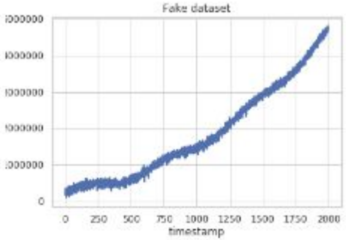
values)

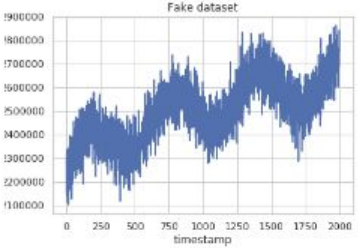


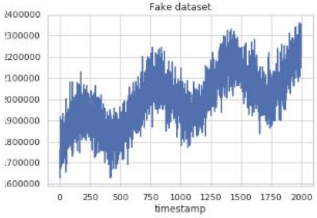


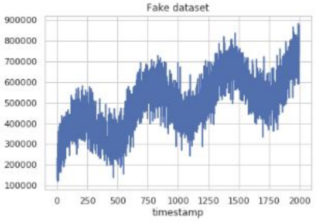


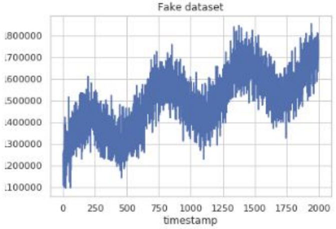


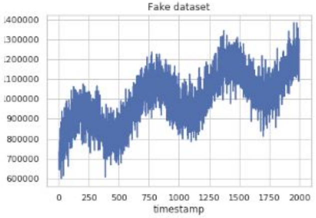








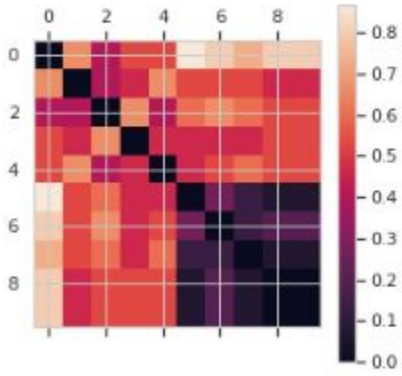




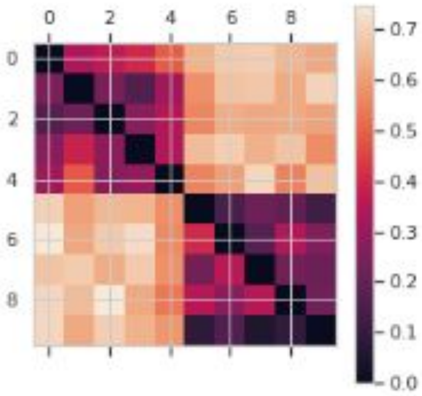
Experiments: Clustering

Fake dataset

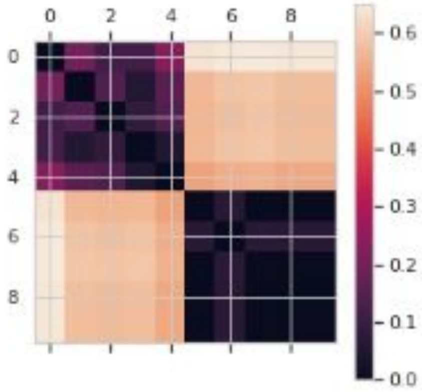
Edit distance:



Label substructure:

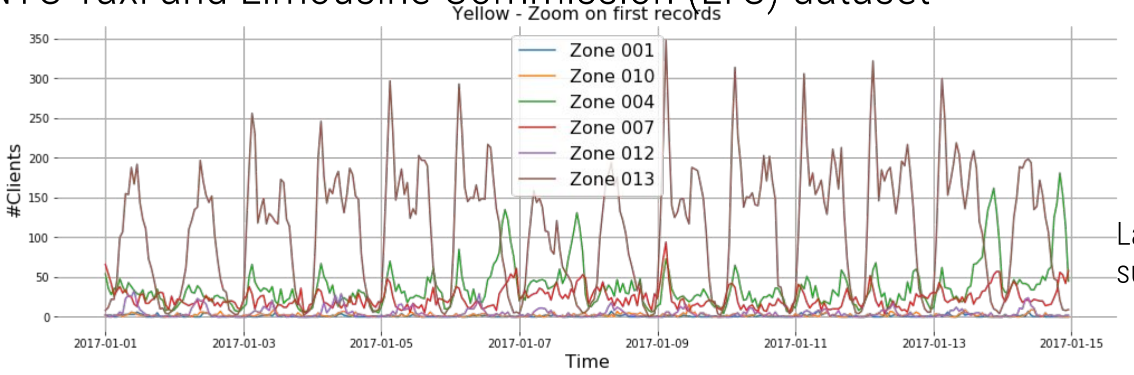


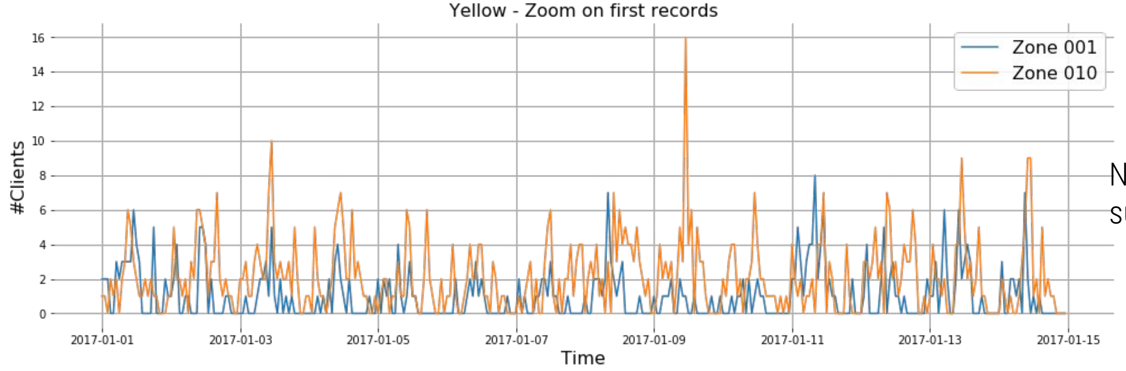
Nonvaule substructure:



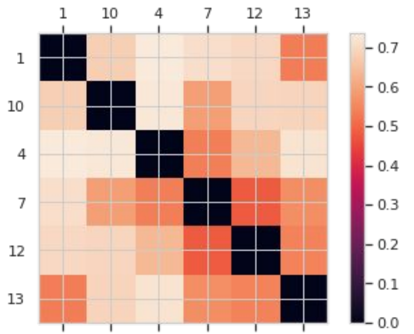
Experiments: Clustering

NYC Taxi and Limousine Commission (LTC) dataset

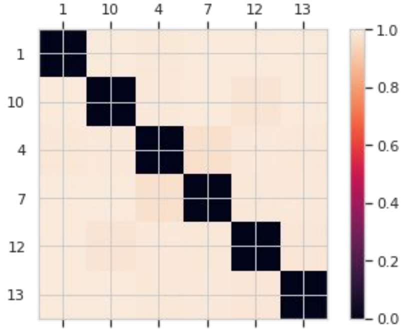




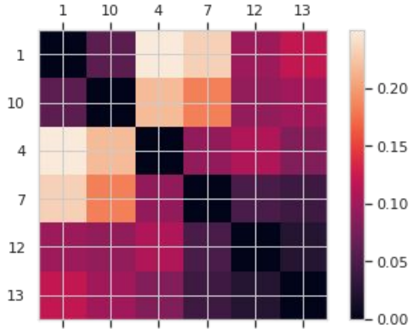
Edit distance:



Label substructure:



Nonvalue substructure:



Experiments: Analysis

Why Label substructure and Edit distance seem to have worse performance?

Maybe because of **overfitting**!

For regression task, the generated DTs usually have huge node number

Experiments: Conclusion

The nonvalue substructure method shows better performance on regression tasks.

craft ai

Q&A

Thank you for your attention!