Summarizing Network Configuration Patterns

Jyotirmay Chauhan, and Aaron Gember-Jacobson

① Network Configurations

Network Configurations: A set of rule/instructions which dictate the flow of packets in a network

Sample Configuration Stanza:

```
interface Coop101
   description Dining Student
   switchport allowed vlans 200

interface Frank101
   description Dining Student
   switchport allowed vlans 200
100

interface McGreg101
   description Student Admin
   switchport allowed vlans None
```

```
interface Case101
   description Library Student
   switchport allowed vlans 100

vlan 100
    name hub

vlan 200
    name backup-hub
```

② Problem Statement

Modern network configuration are huge and extremely complex. Challenge from a debugging
Perspective. One method for this is **Model checking**:

Pros of Model Checking:- ***Highly*** Accurate error checking,
Cons of Model Checking:- ***Difficulty*** in Model creation
Difficulty in Specification enumeration

Alternate Strategy: Infer patterns from network configurations
Existing research: Minerals [7] and SelfStarter [6] infer patterns about the Interfaces, ACLs and/or BGP instances but, ignore layer-2 components, syntactic sugar, and comments.

Our Approach: Identify significant and useful difference between different network configurations.

③ Contrast Set Learning

Contrast Set Learning identifies **meaningful differences** between separate groups

Relationships between components can be viewed as a set of **IF-THEN rules**

Eg: **Iface** is Anchor component *(Primary-key)*
vlan100, vlav200, vlan300 etc are associated components

Ifaces	vlan100	vlan200	vlan300	Student	Dining
case101	1	0	0	1	0
case102	0	1	0	1	1
coop101	0	1	0	1	1
frank101	1	1	0	1	1
mcg101	0	0	1	1	1

Contrast set: conjunction of known attribute-value pairs
IF : vlan100 = 1 & vlan200 = 1 ***(rule length 2)***

Group feature: attribute-value pair we are trying to predict
THEN : Iface = frank101

④ Rule Pruning

Rules generated by CSL algorithm STUCCO: 1~2 million (for a Uni-size dataset)
Necessitates **Rule Filtering.**

Existing metrics:-

- Precision : High Precision
- Recall : Not a useful metric
  - Observed evidence: Low Recall rule CAN be useful
- Frequency: Count is relevant

Our Metric:-

- **Rule Coverage:**
  - Rows where the **IF part + THEN part** of rule is **satisfied**
  - Number of such rows = **Impact of Rule**

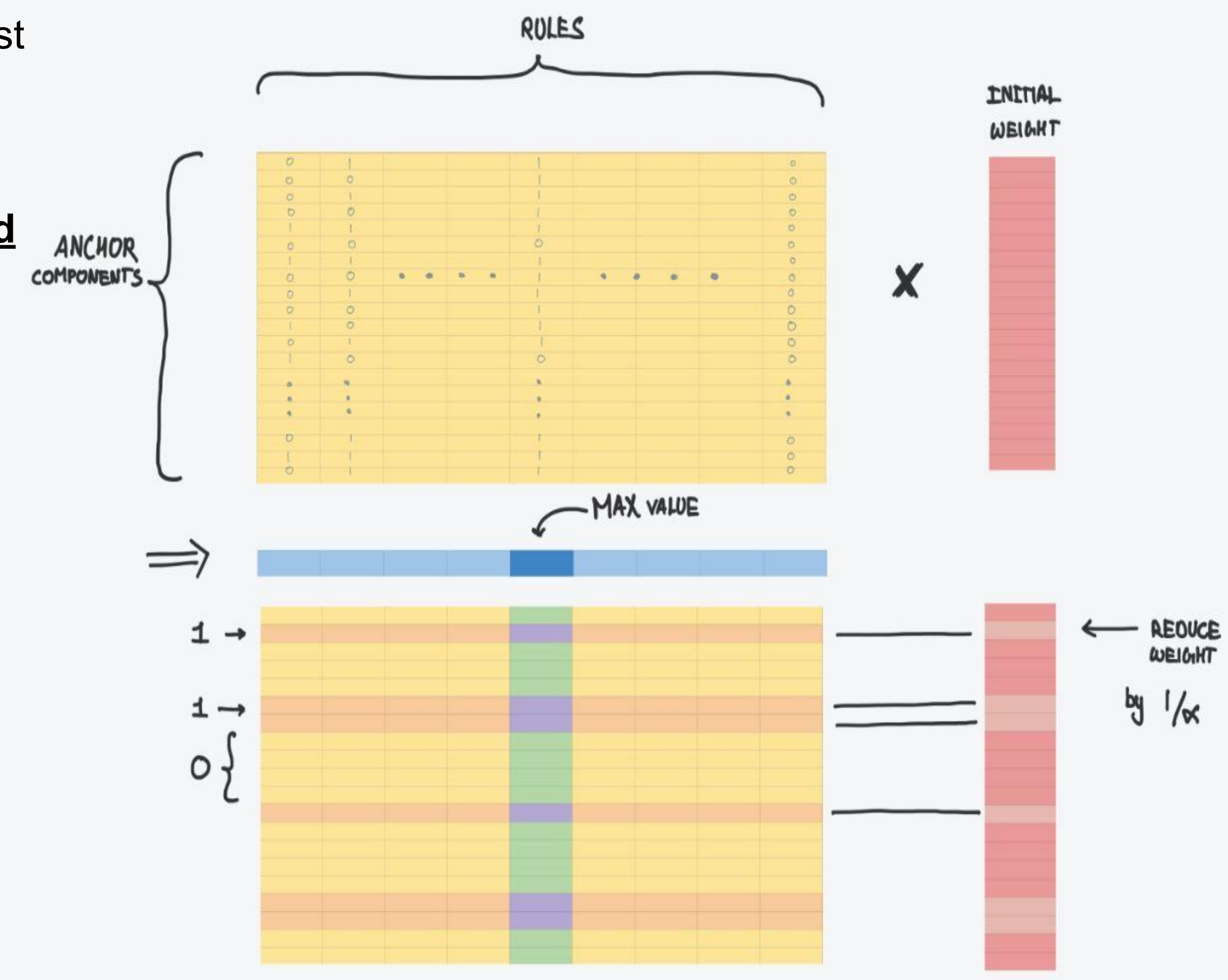
⑤ Rule-Set Cover

A Greedy Heuristic:
Isolates the most important rules for each group-feature into **Condensed Rule-Set**

Step 1

MAX value **RULE** saved to Output Rule-set

Change **Weights**

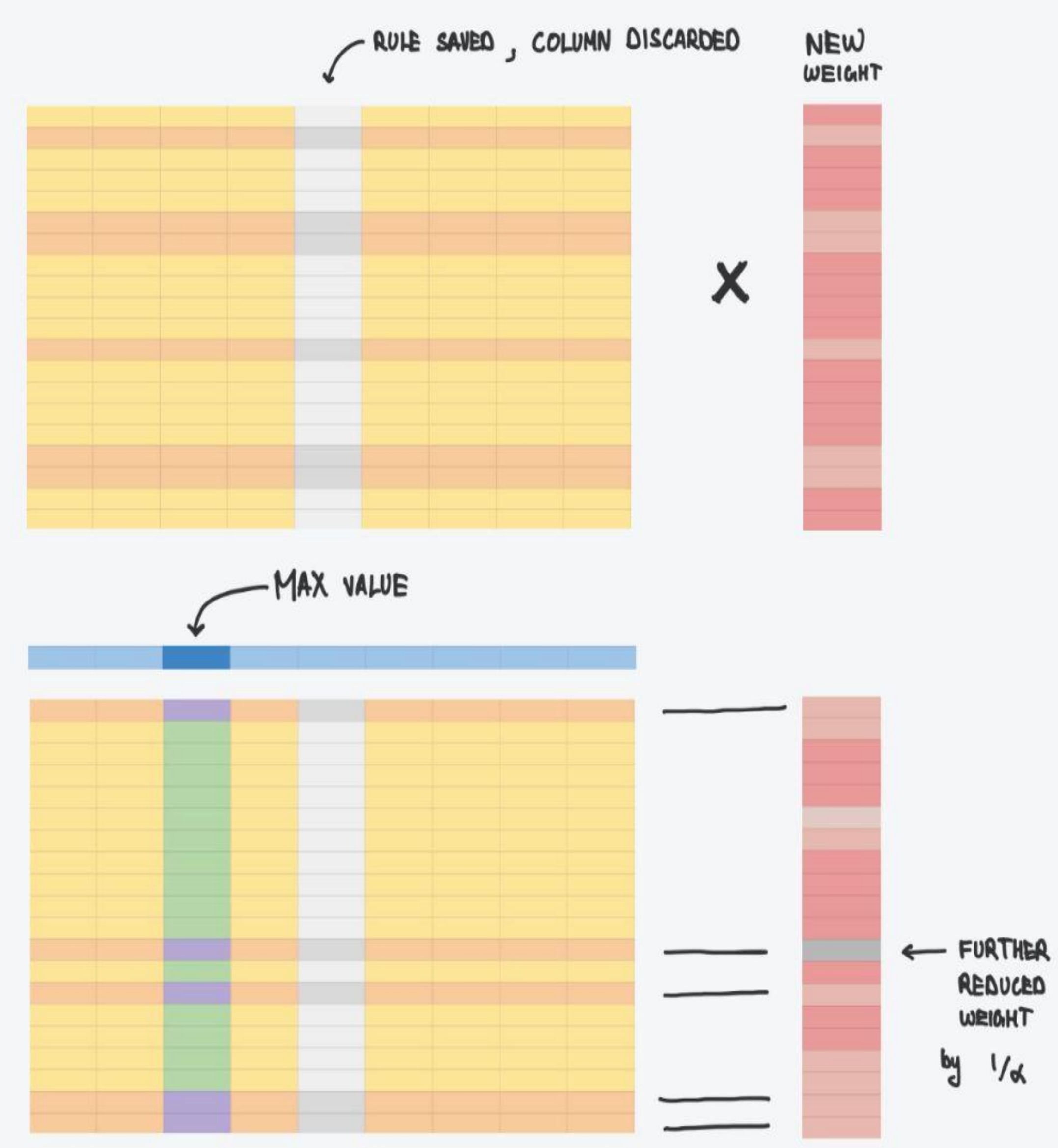


Step 2

Discard previously selected column

Find new MAX **RULE**

Repeat



⑥ Rule-Set Summarization

Problem with Rules in Output Rule-set: STILL LARGE-
Superior rules often exist.
Superior rule:

- Shorter length but SIMILAR ***Precision*** & ***Rule Coverage***

Solution: Rule-set Condensation

Idea: EXTRACT Common Elements

Rule A: *IF : vlan100 = 1 & vlan300= 0* *THEN* Student = 1
Rule B: *IF : vlan100 = 1* *THEN* Student = 1

Rule B **Superior**!
Common Element: *vlan100 = 1*

Acknowledgements & References

- **F. Le, S. Lee, T. Wong, H. S. Kim, and D. Newcomb. Detecting network-wide and router-specific misconfigurations through data mining. IEEE/ACM Trans. Netw., 17(1):66–79, 2009**
- **S. D. Bay and M. J. Pazzani. Detecting change in categorical data: Mining contrast sets. In KDD, 1999.**