# Ensemble Trees

## Leveraging Ensemble Power inside Decision Trees

Discovery Science 2008
Budapest, 14.10.2008

Albrecht Zimmermann
Katholieke Universiteit Leuven

# The Setting

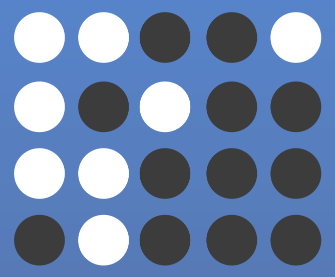
- Vectorial data $\langle v_1, v_2, v_3, \ldots, v_d \rangle$
- Binary class {pos,neg}

# Motivation

Decision trees easy to induce

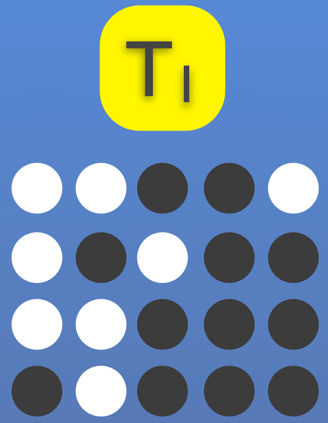
# Motivation

Decision trees easy to induce



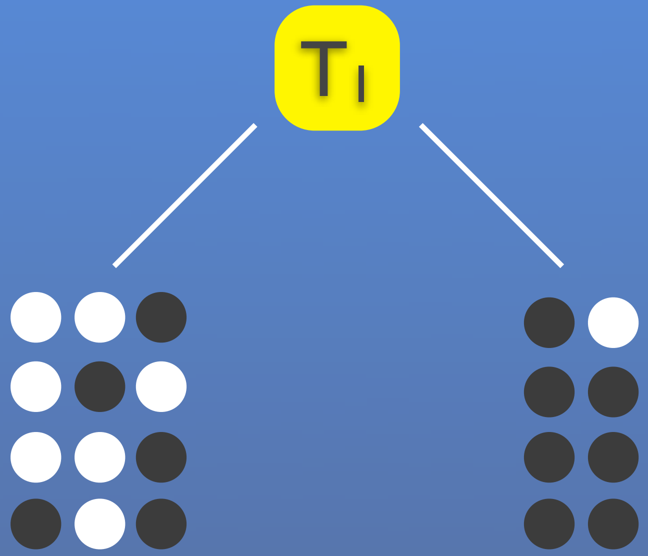
# Motivation

Decision trees easy to induce



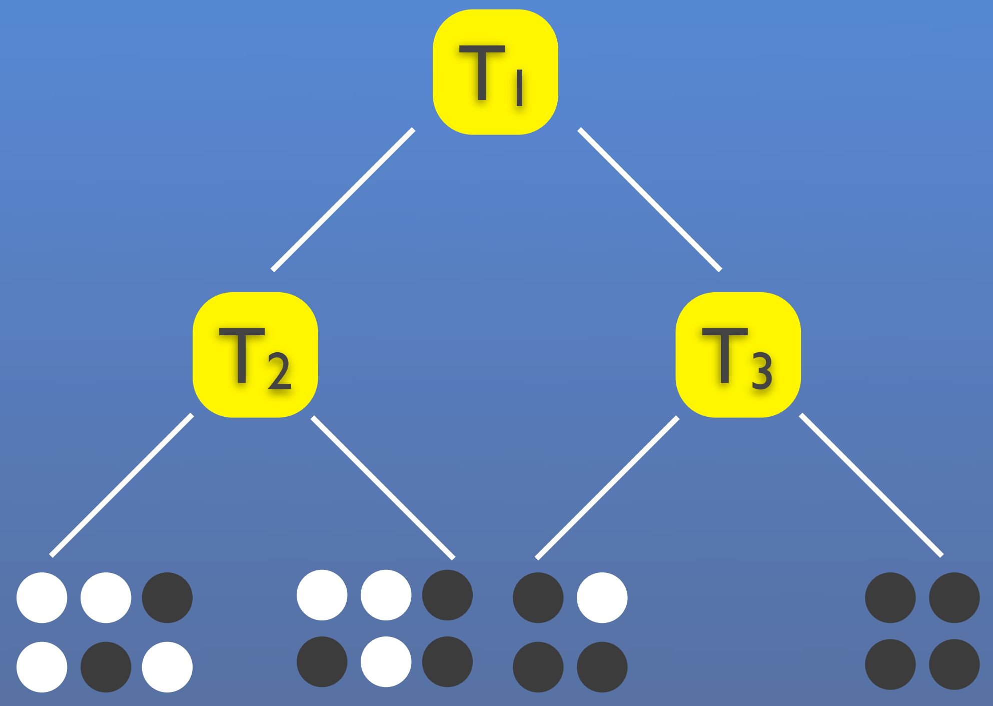
# Motivation

Decision trees easy to induce



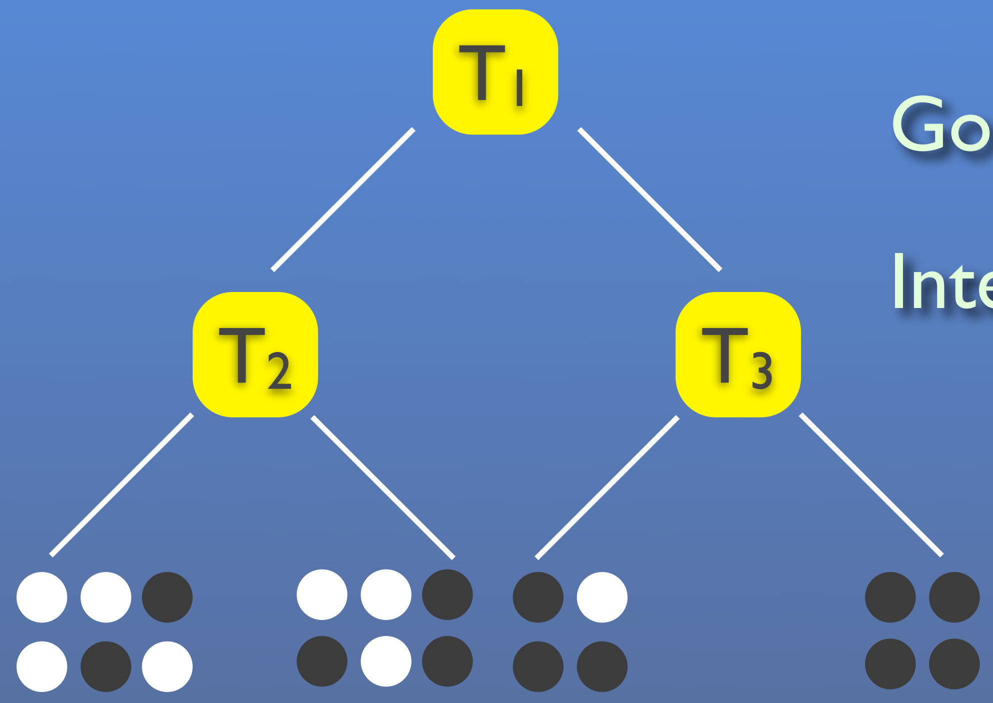
# Motivation

Decision trees easy to induce



# Motivation

Decision trees easy to induce

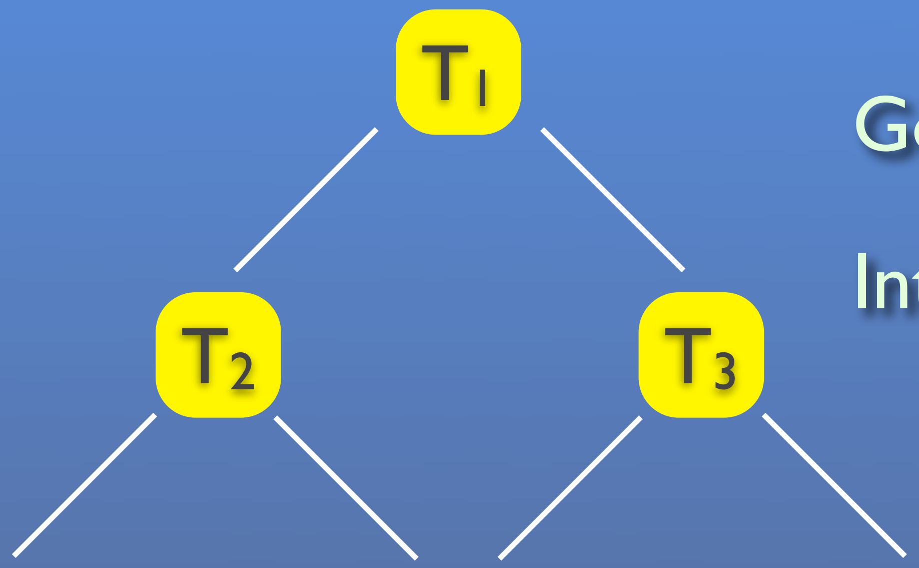


Good accuracy ✓

Interpretable ✓

# Motivation

Decision trees easy to induce



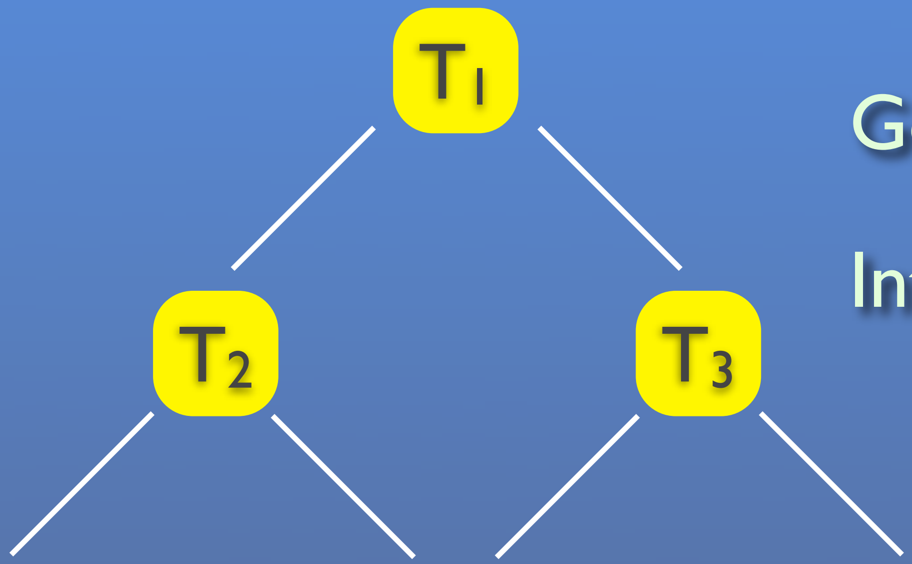
Good accuracy ✓

Interpretable ✓

Problem: Changes in the Data

# Motivation

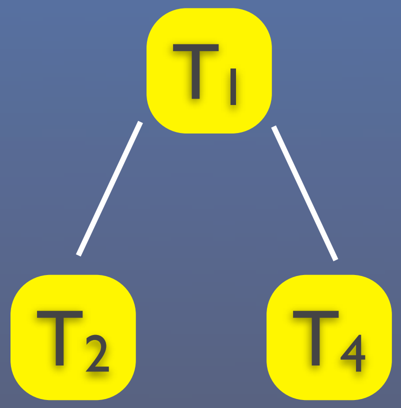
Decision trees easy to induce



Good accuracy ✓

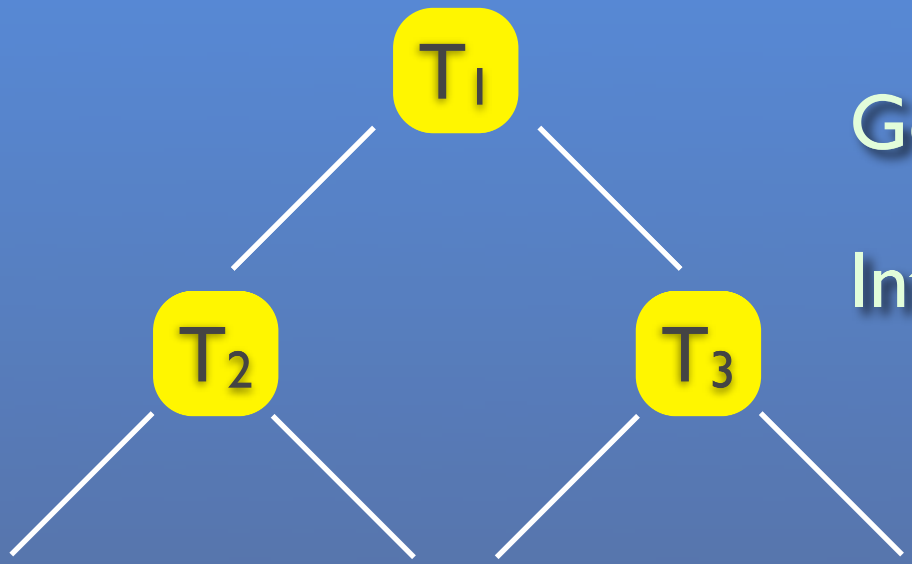
Interpretable ✓

Problem: Changes in the Data



# Motivation

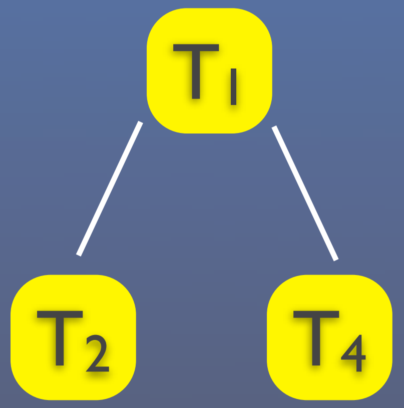
Decision trees easy to induce



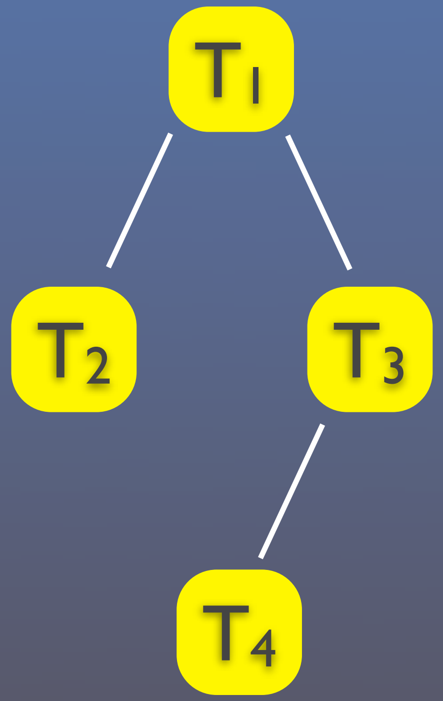
Good accuracy ✓

Interpretable ✓

Problem: Changes in the Data

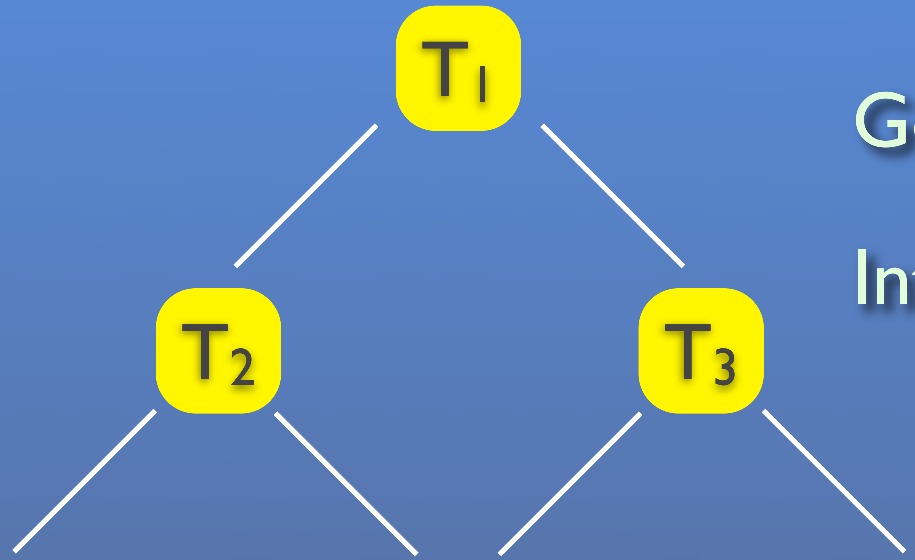


or



# Motivation

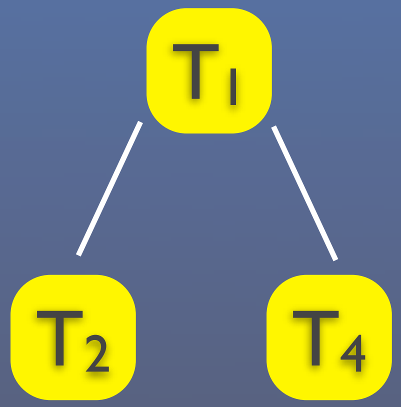
Decision trees easy to induce



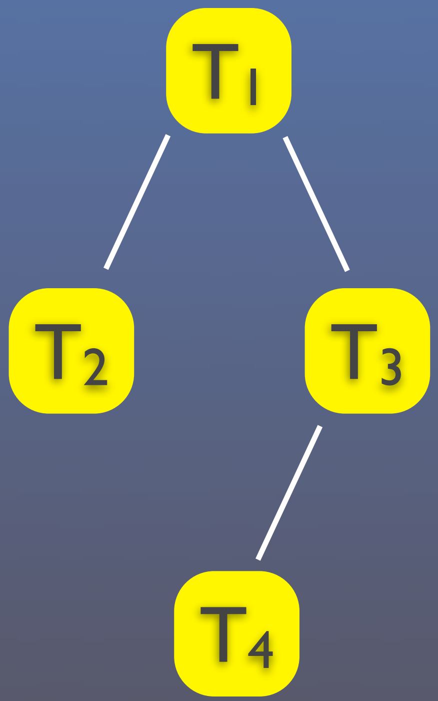
Good accuracy ✓

Interpretable ✓

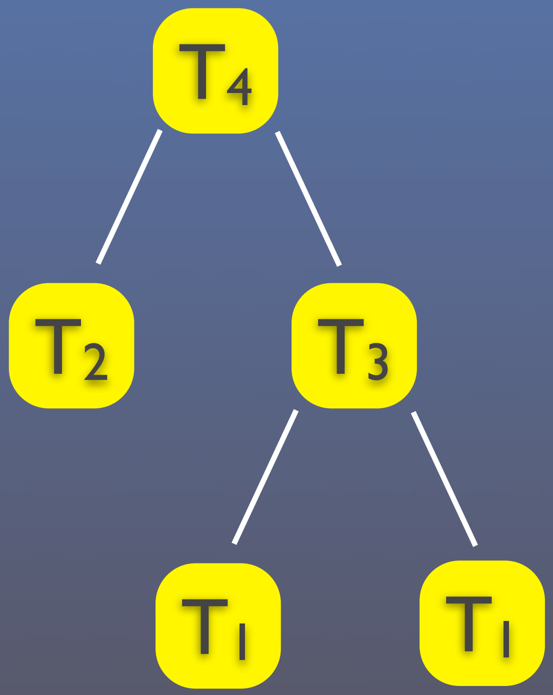
Problem: Changes in the Data



or

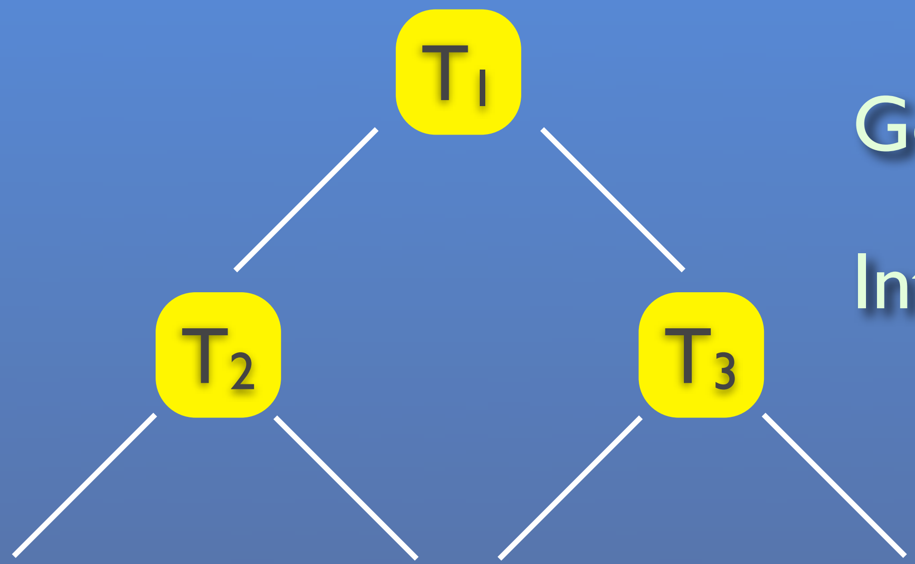


or even



# Motivation

Decision trees easy to induce

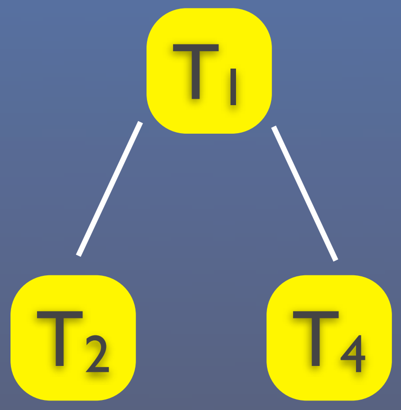


Good accuracy ✓

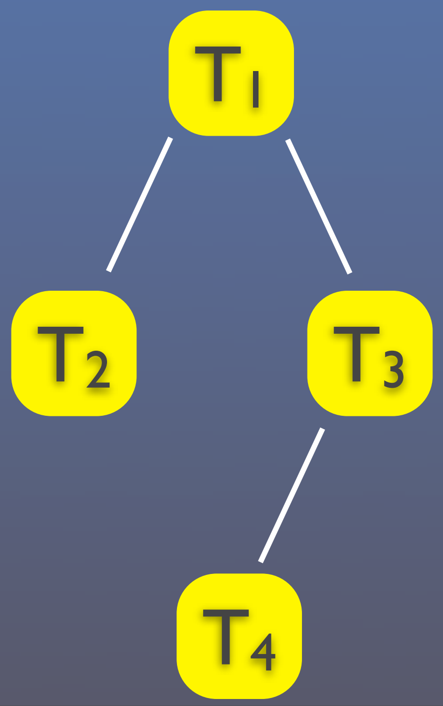
Interpretable ✓

Varying accuracies ✗
Faulty interpretations ✗

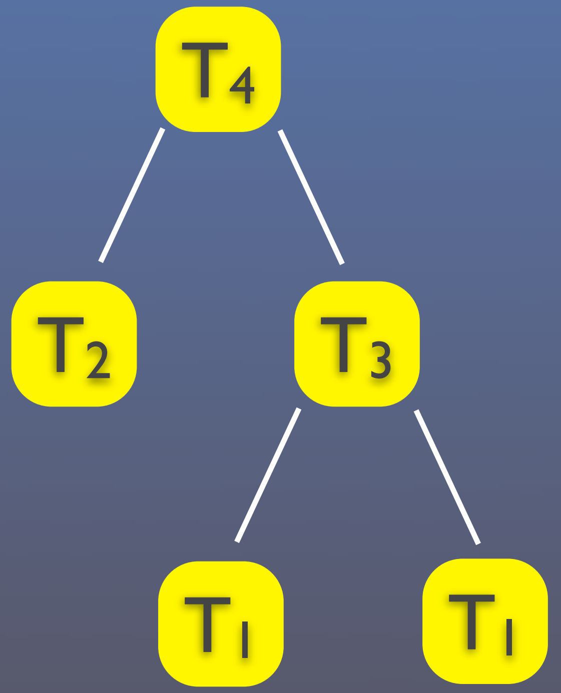
Problem: Changes in the Data



or



or even

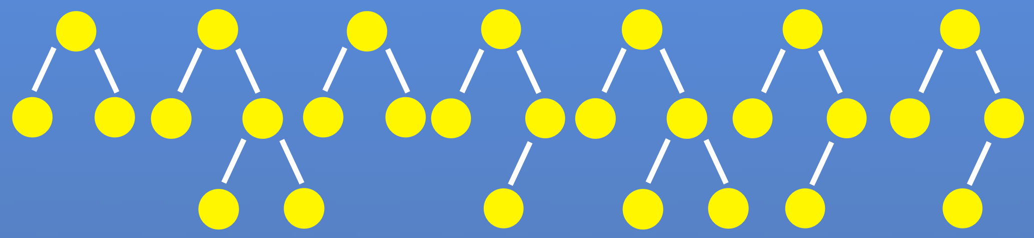


# Existing Solutions

Solution 1: lots of trees!

# Existing Solutions

Solution 1: lots of trees!

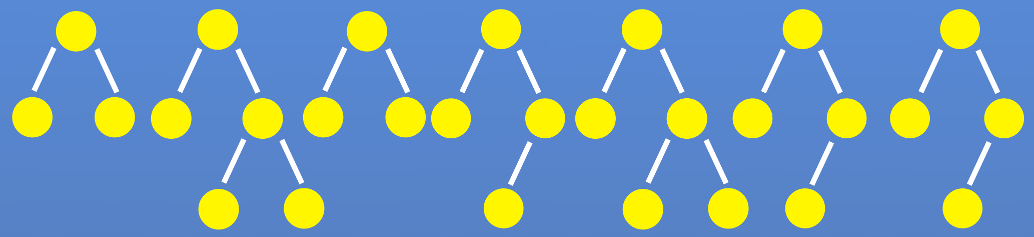


Ensembles:

- Better accuracy
- Essentially black boxes
- Stability N/A

# Existing Solutions

Solution 1: lots of trees!

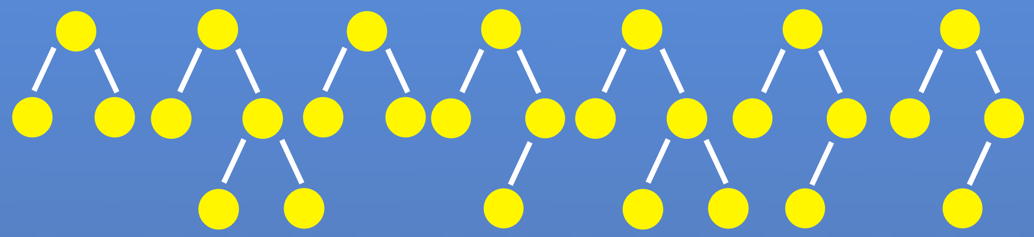


Ensembles:

- Better accuracy ✓
- Essentially black boxes
- Stability N/A

# Existing Solutions

Solution 1: lots of trees!

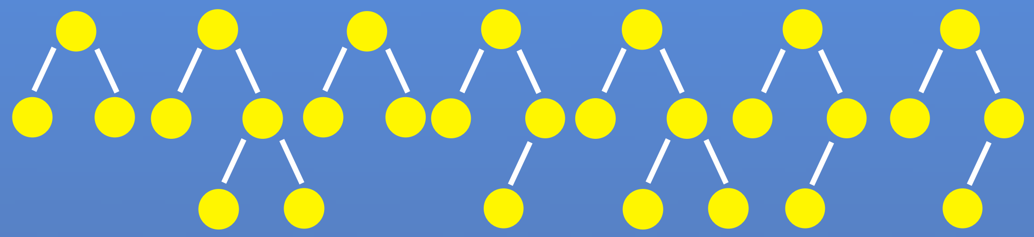


Ensembles:

- Better accuracy ✓
- Essentially black boxes ✗
- Stability N/A

# Existing Solutions

Solution 1: lots of trees!



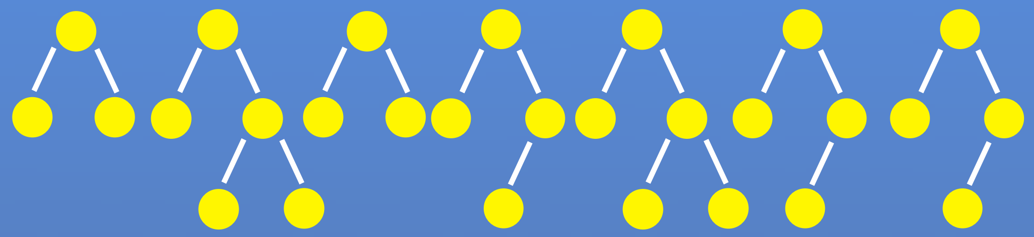
Solution 2: more expressive tests

Ensembles:

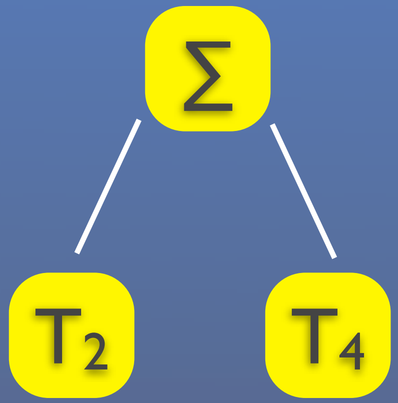
- Better accuracy ✓
- Essentially black boxes ✗
- Stability N/A

# Existing Solutions

Solution 1: lots of trees!



Solution 2: more expressive tests



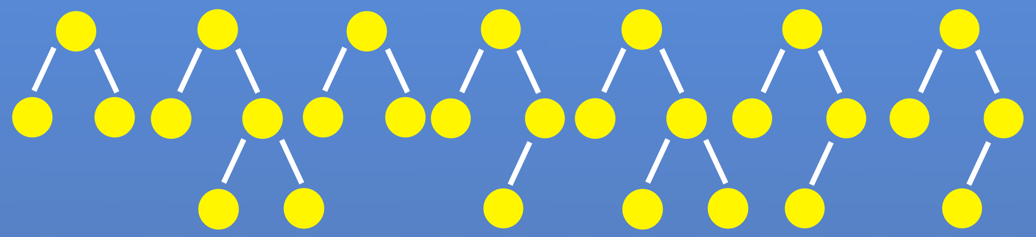
(Linear) combinations

Ensembles:

- Better accuracy ✓
- Essentially black boxes ✗
- Stability N/A

# Existing Solutions

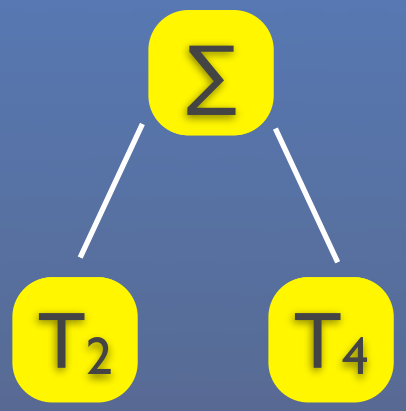
Solution 1: lots of trees!



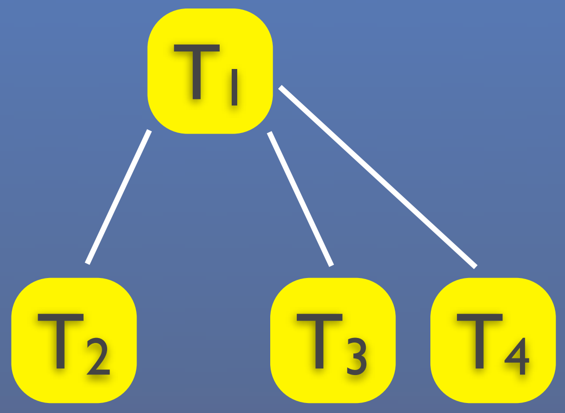
Ensembles:

- Better accuracy ✓
- Essentially black boxes ✗
- Stability N/A

Solution 2: more expressive tests



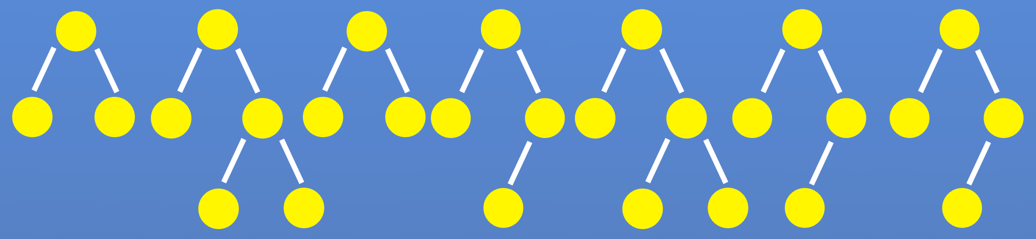
(Linear) combinations



Option Trees

# Existing Solutions

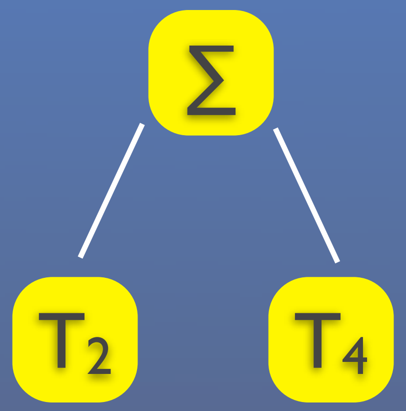
Solution 1: lots of trees!



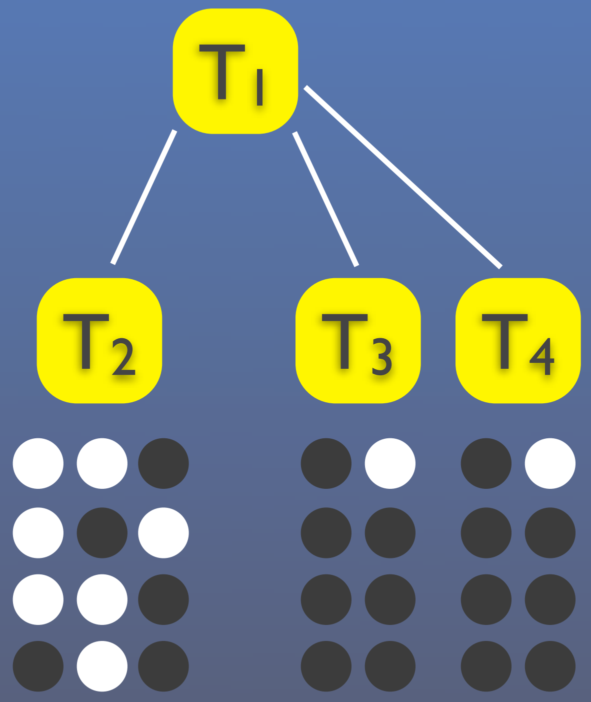
Ensembles:

- Better accuracy ✓
- Essentially black boxes ✗
- Stability N/A

Solution 2: more expressive tests



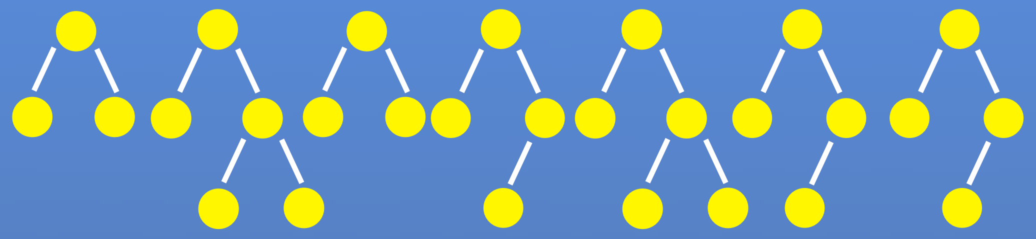
(Linear) combinations



Option Trees

# Existing Solutions

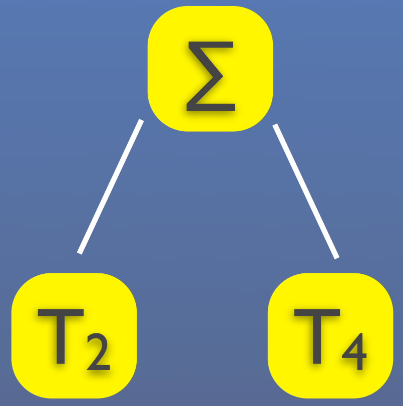
Solution 1: lots of trees!



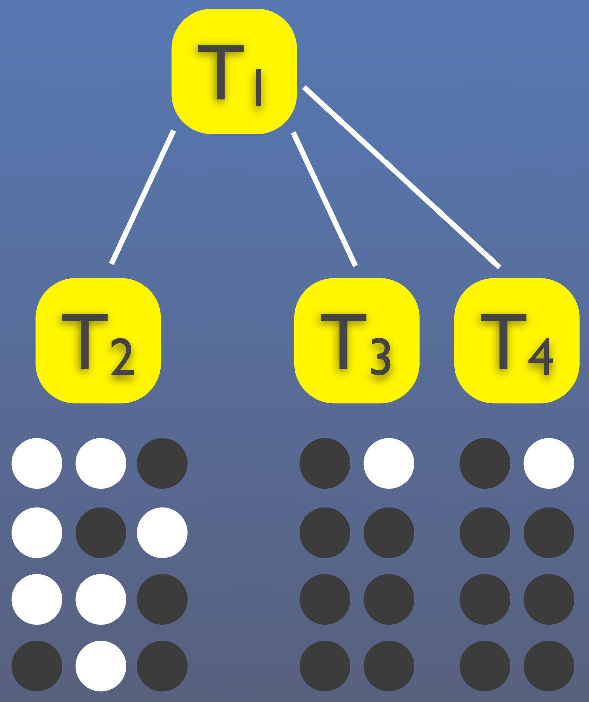
Ensembles:

- Better accuracy ✓
- Essentially black boxes ✗
- Stability N/A

Solution 2: more expressive tests



(Linear) combinations



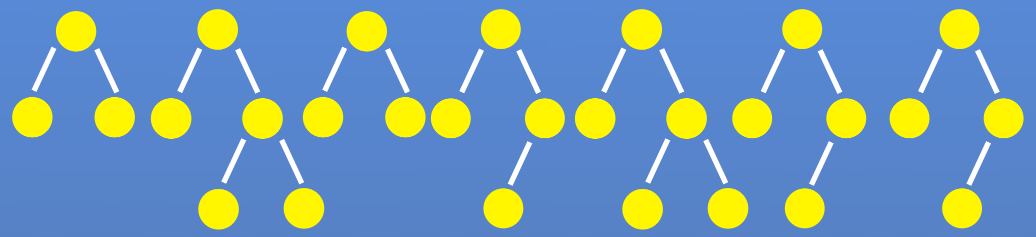
Option Trees

Complex tests

- Better accuracy
- Size OTs = size ensembles
- Complex tests opaque
- Stability?

# Existing Solutions

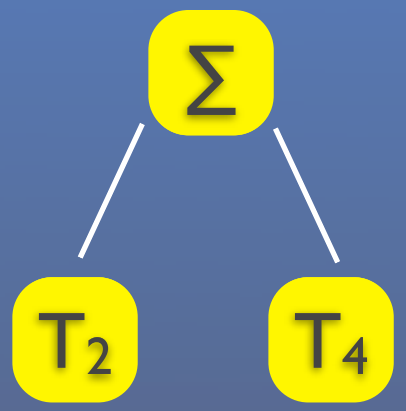
Solution 1: lots of trees!



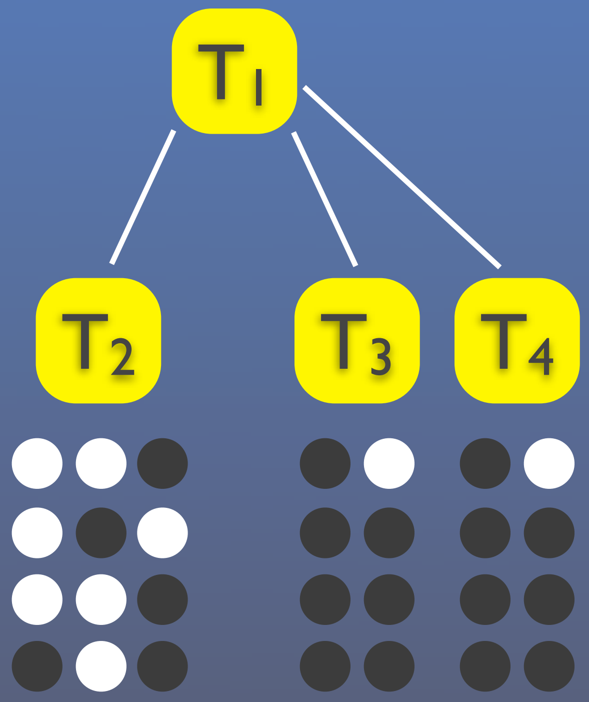
Ensembles:

- Better accuracy ✓
- Essentially black boxes ✗
- Stability N/A

Solution 2: more expressive tests



(Linear) combinations



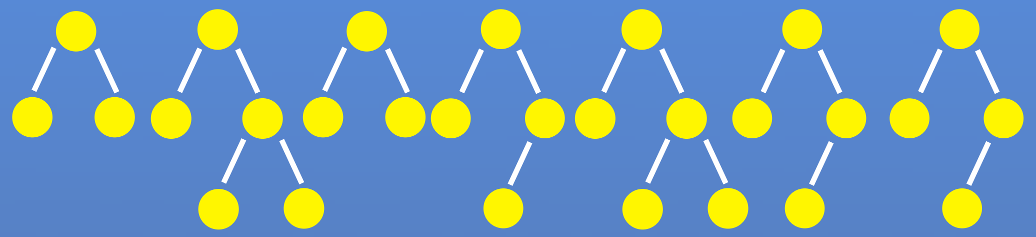
Option Trees

Complex tests

- Better accuracy ✓
- Size OTs = size ensembles
- Complex tests opaque
- Stability?

# Existing Solutions

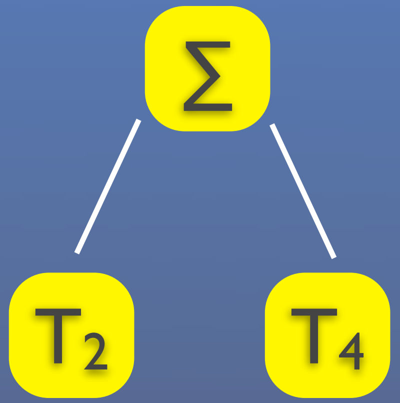
Solution 1: lots of trees!



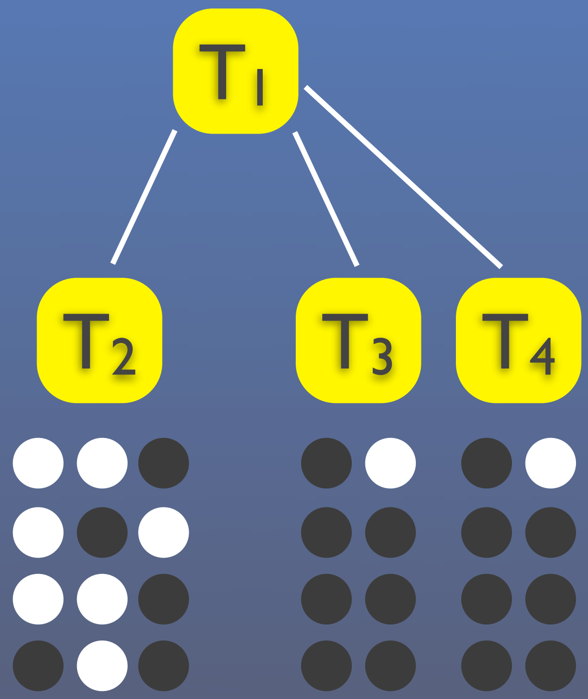
Ensembles:

- Better accuracy ✓
- Essentially black boxes ✗
- Stability N/A

Solution 2: more expressive tests



(Linear) combinations



Option Trees

Complex tests

- Better accuracy ✓
- Size OTs = size ensembles ✗
- Complex tests opaque ✗
- Stability?

# Goals

G1: High, consistent accuracy

G2: Small total number of nodes

G3: Structural stability w.r.t. data changes

# Our Solution

Accuracy & stability:

- Ensembles as tests
- Statistically quantified patterns

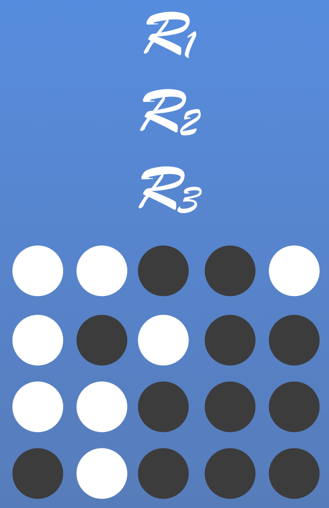
Size/Interpretability:

- Keep ensembles small
- Conjunctive patterns

# Algorithmic aspects

# Algorithmic aspects

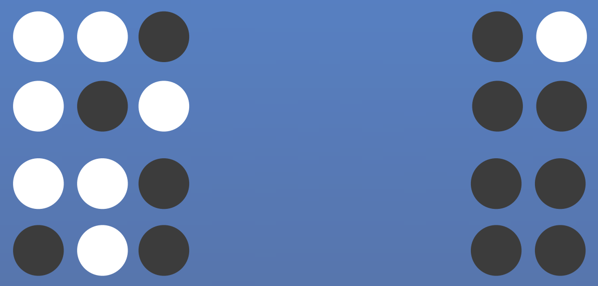
1. Induce *k* best patterns (IG)



# Algorithmic aspects

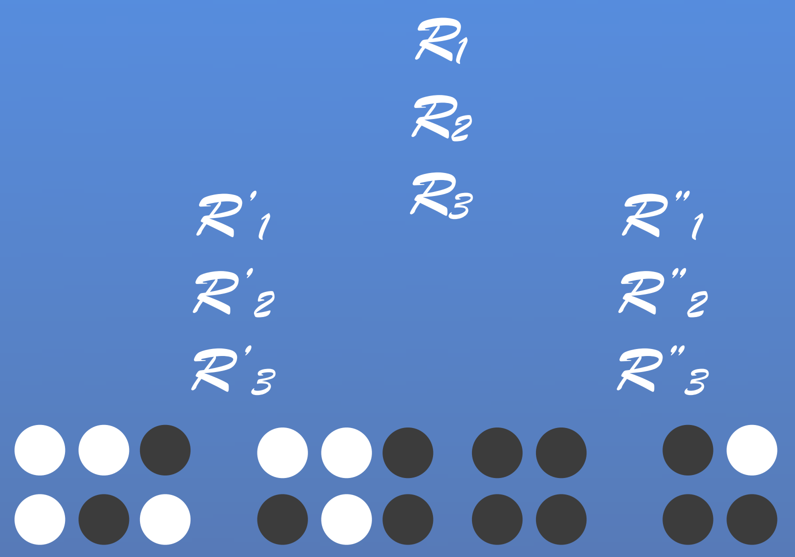
1. Induce *k* best patterns (IG)
2. Split data

$\mathcal{R}_1$
$\mathcal{R}_2$
$\mathcal{R}_3$



# Algorithmic aspects

1. Induce *k* best patterns (IG)
2. Split data
3. Repeat until
   - Subset too small
   - Less than *k* rules found



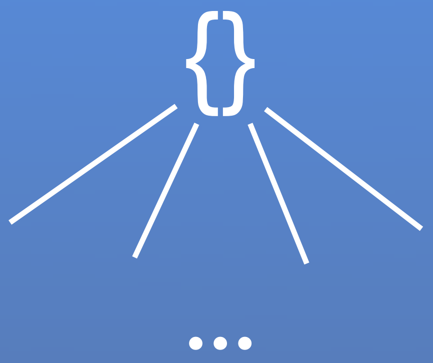
# Finding The Patterns

Branch-and-bound search:

# Finding The Patterns

Branch-and-bound search:

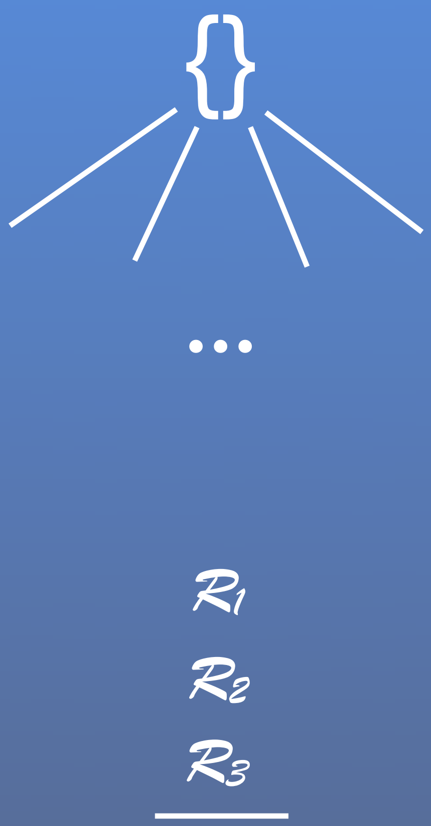
- Enumerate/evaluate pattern(s)



# Finding The Patterns

Branch-and-bound search:

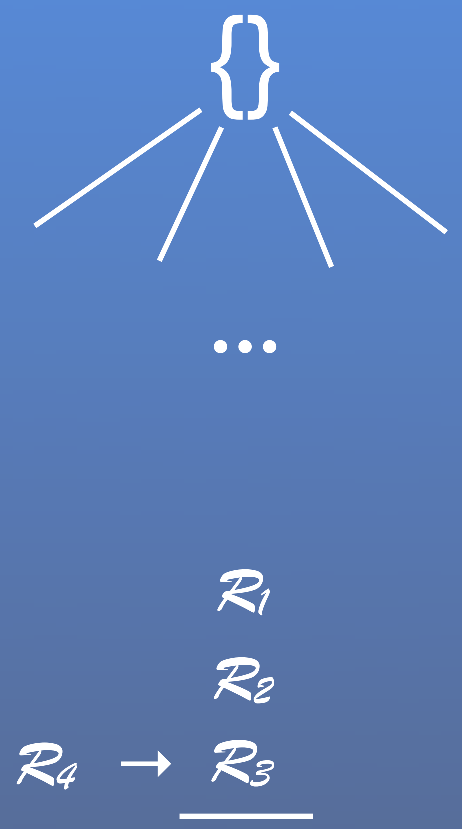
- Enumerate/evaluate pattern(s)



# Finding The Patterns

Branch-and-bound search:

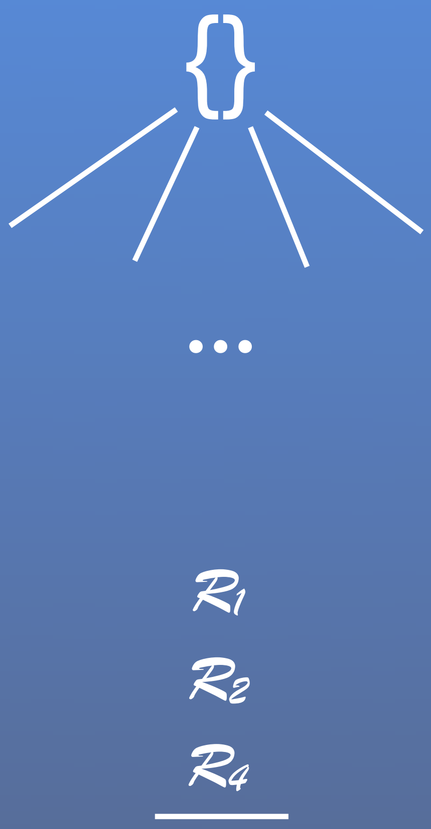
- Enumerate/evaluate pattern(s)
- Better than current *k*th-best, keep



# Finding The Patterns

Branch-and-bound search:

- Enumerate/evaluate pattern(s)
- Better than current *k*th-best, keep



{}

...

$R_1$

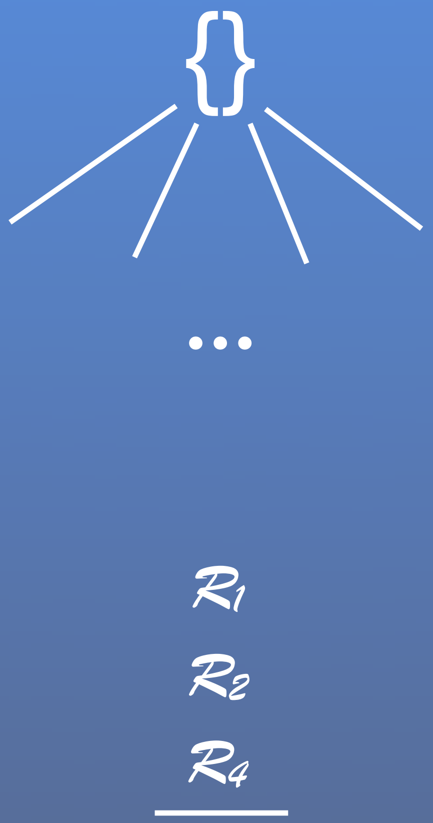
$R_2$

$R_4$

# Finding The Patterns

Branch-and-bound search:

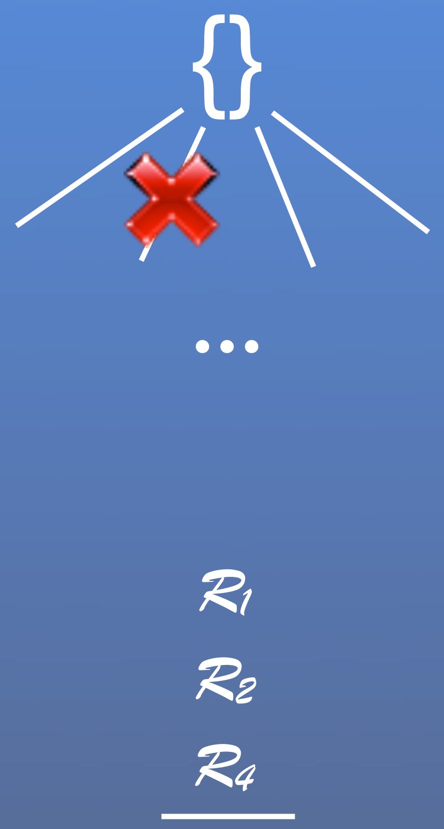
- Enumerate/evaluate pattern(s)
- Better than current *k*th-best, keep
- Worse than current *k*th-best, discard



# Finding The Patterns

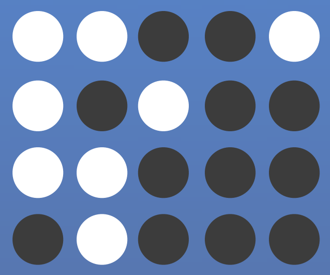
Branch-and-bound search:

- Enumerate/evaluate pattern(s)
- Better than current *k*th-best, keep
- Worse than current *k*th-best, discard
- Prune using upper bounds



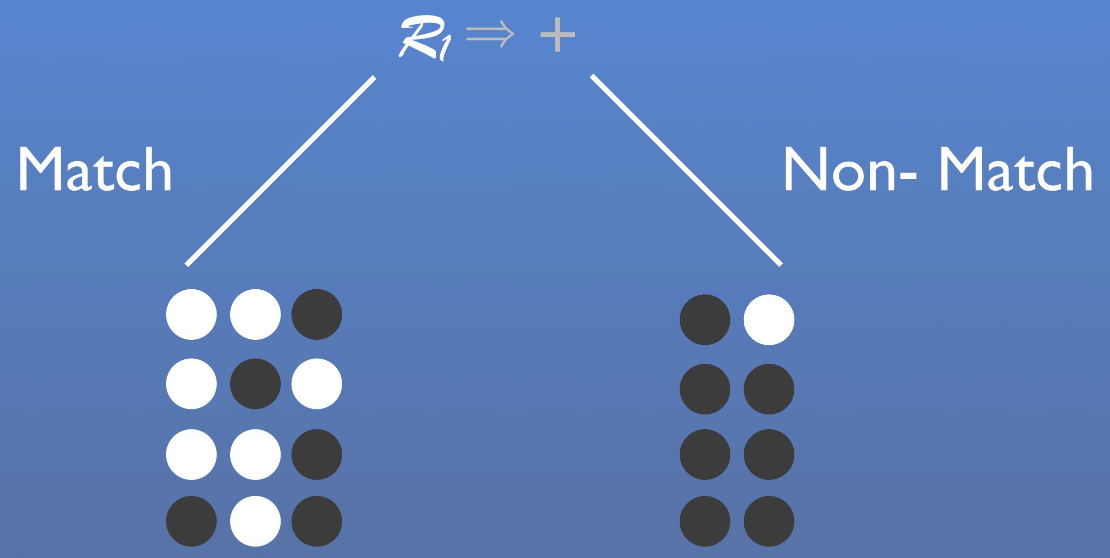
# Splitting Data

One Pattern: simple, match/non-match



# Splitting Data

## One Pattern: simple, match/non-match



$\mathcal{R}_1 \Rightarrow +$

Match

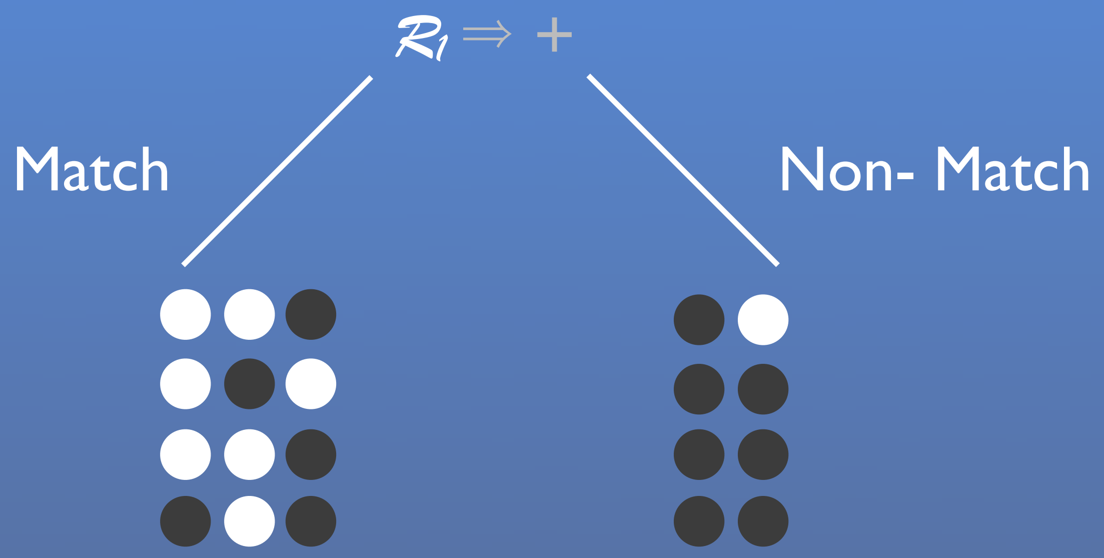
Non- Match

# Splitting Data

Several Patterns: less simple

# Splitting Data

## Several Patterns: less simple



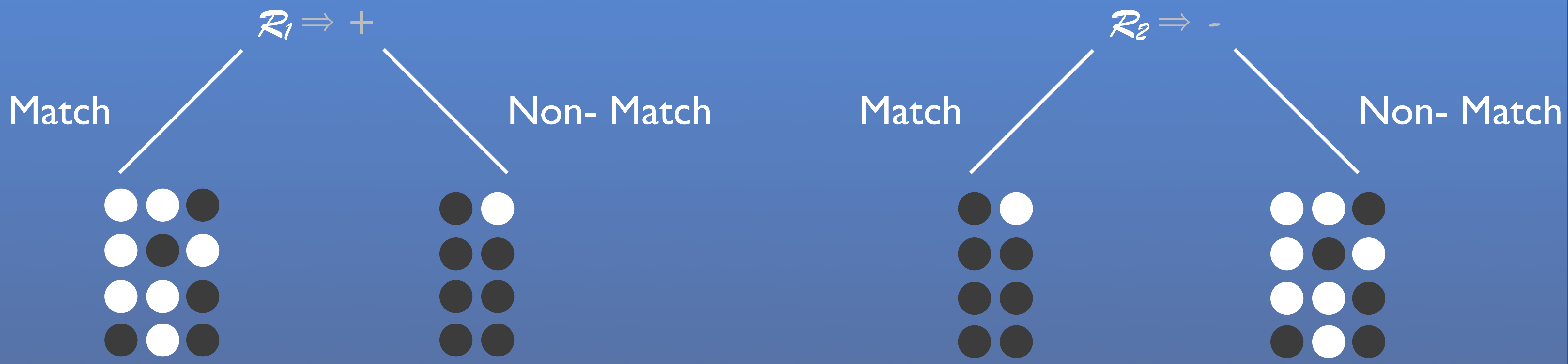
$\mathcal{R}_1 \Rightarrow +$

Match

Non- Match

# Splitting Data

## Several Patterns: less simple



$\mathcal{R}_1 \Rightarrow +$

Match

Non- Match

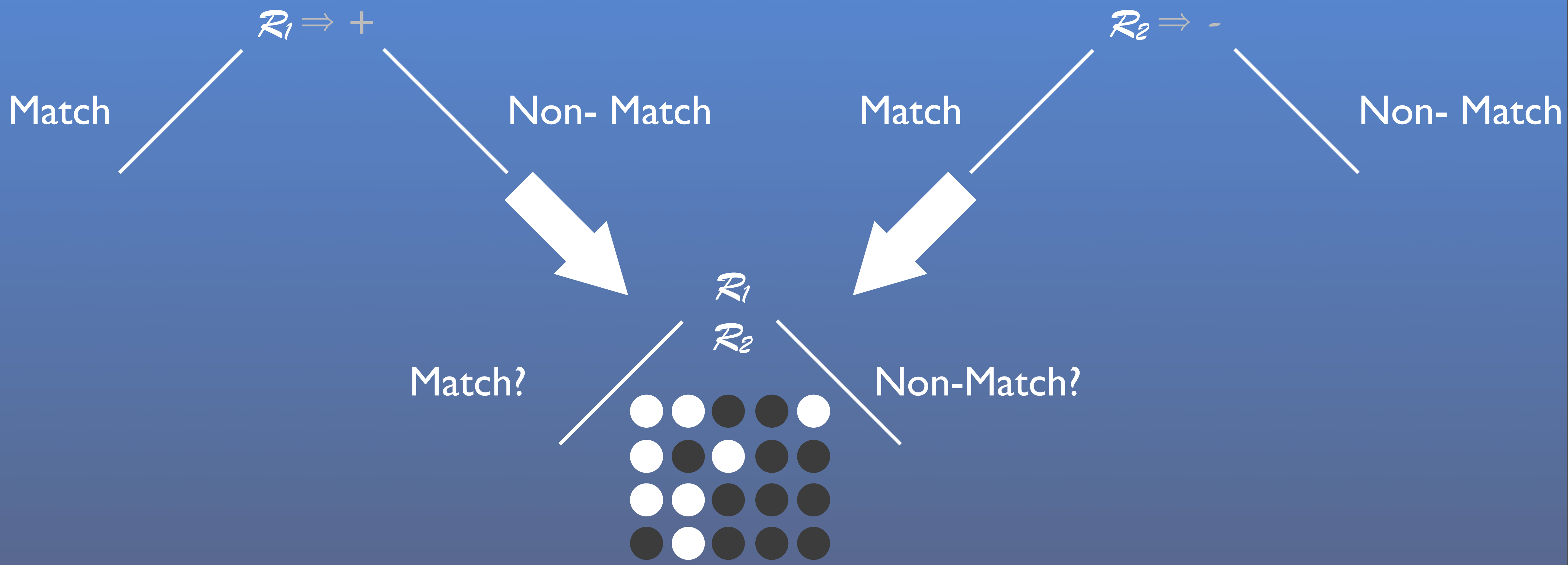
$\mathcal{R}_2 \Rightarrow -$

Match

Non- Match

# Splitting Data

## Several Patterns: less simple



$\mathcal{R}_1 \Rightarrow +$

Match

Non- Match

$\mathcal{R}_2 \Rightarrow -$

Match

Non- Match

$\mathcal{R}_1$

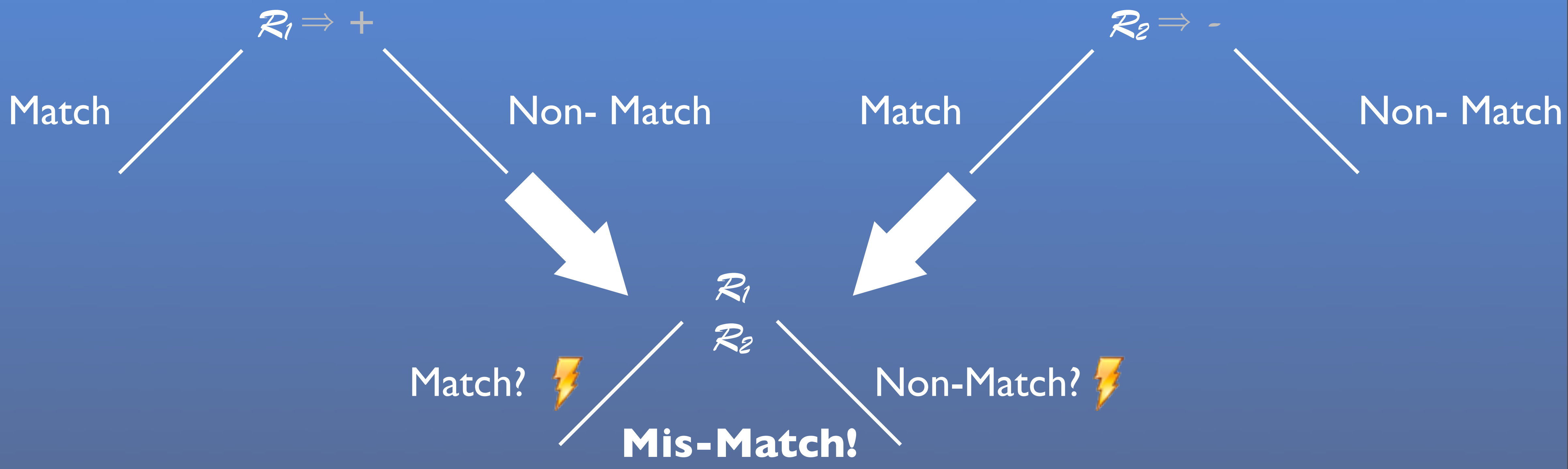
$\mathcal{R}_2$

Match?

Non-Match?

# Splitting Data

## Several Patterns: less simple



$\mathcal{R}_1 \Rightarrow +$

Match

Non- Match

$\mathcal{R}_2 \Rightarrow -$

Match

Non- Match

$\mathcal{R}_1$

$\mathcal{R}_2$

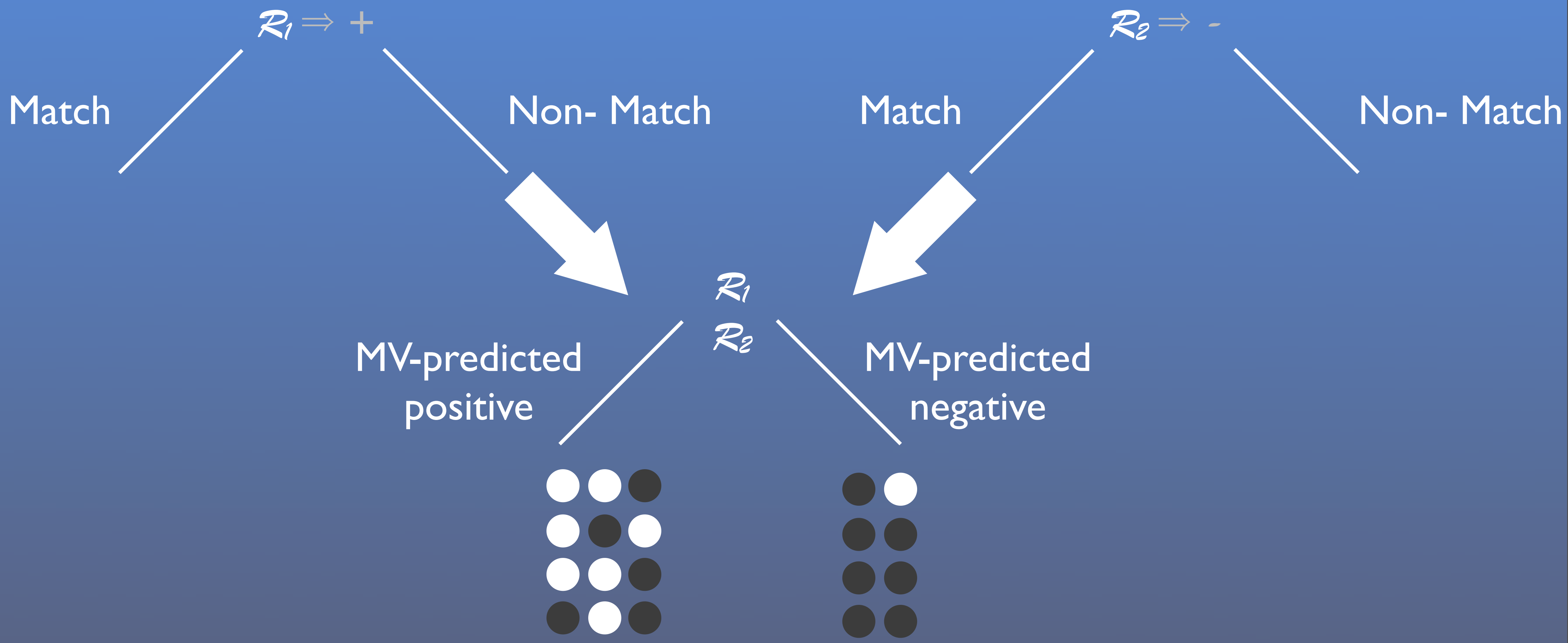
Match? ⚡

Non-Match? ⚡

**Mis-Match!**

# Splitting Data

## Several Patterns: less simple



$\mathcal{R}_1 \Rightarrow +$

Match

Non- Match

$\mathcal{R}_2 \Rightarrow -$

Match

Non- Match

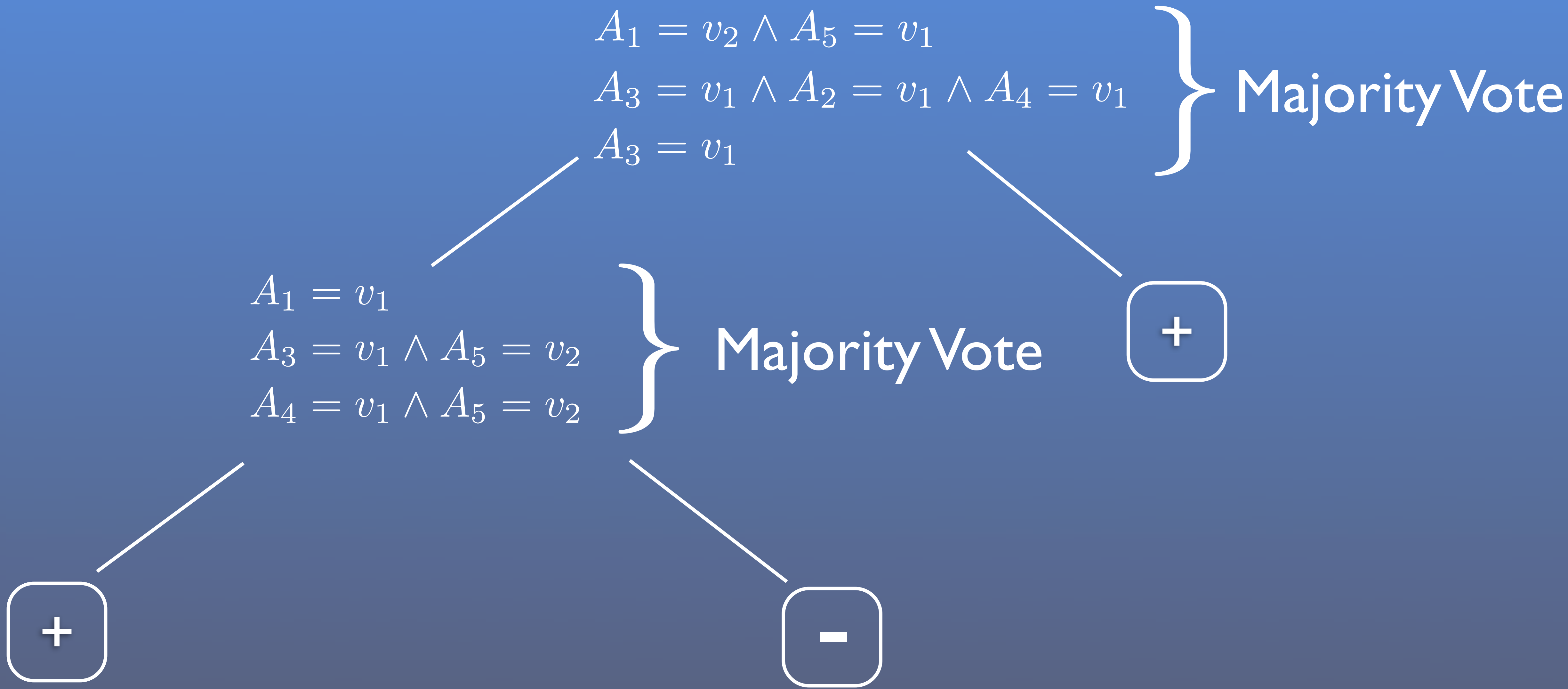
$\mathcal{R}_1$

$\mathcal{R}_2$

MV-predicted positive

MV-predicted negative

# So How Does One Look Like?



$$\left.\begin{array}{l} A_1 = v_2 \wedge A_5 = v_1 \\ A_3 = v_1 \wedge A_2 = v_1 \wedge A_4 = v_1 \\ A_3 = v_1 \end{array}\right\}$$ Majority Vote

$$\left.\begin{array}{l} A_1 = v_1 \\ A_3 = v_1 \wedge A_5 = v_2 \\ A_4 = v_1 \wedge A_5 = v_2 \end{array}\right\}$$ Majority Vote

+

+

-

# Performance

- Compared to Bagging, Boosting, C4.5
- Good accuracy ✓
- Lot smaller than ensembles ✓
  - 1-2 orders of magnitude (# nodes)
- Still as unstable as C4.5, depending on data ✗

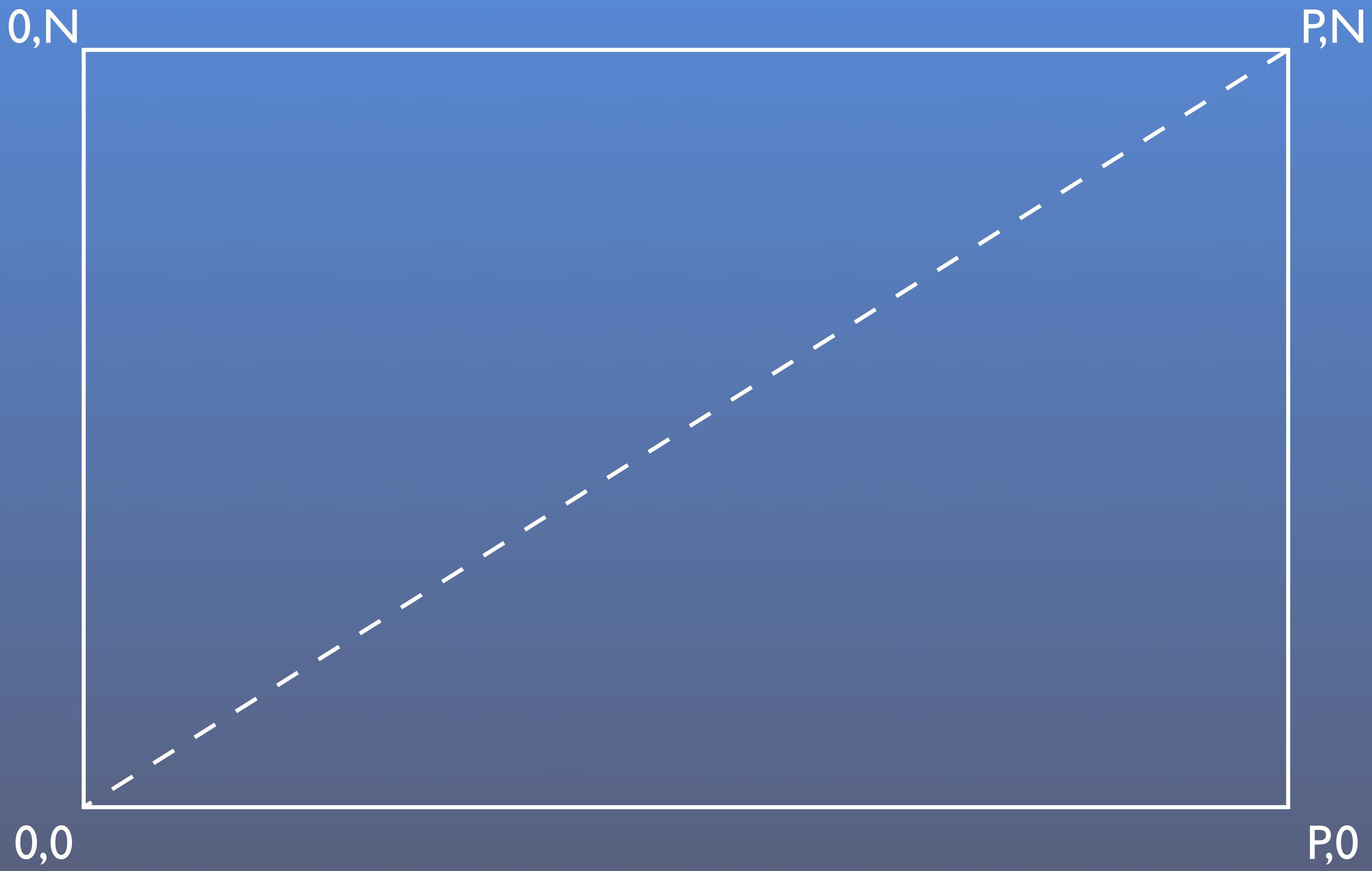
# Summary

- Goal: improve accuracy, stability of DTs
- Invert ensemble process ⇒ inside nodes
- Simple to induce, effective pre-pruning
- Improvements necessary

# But wait...

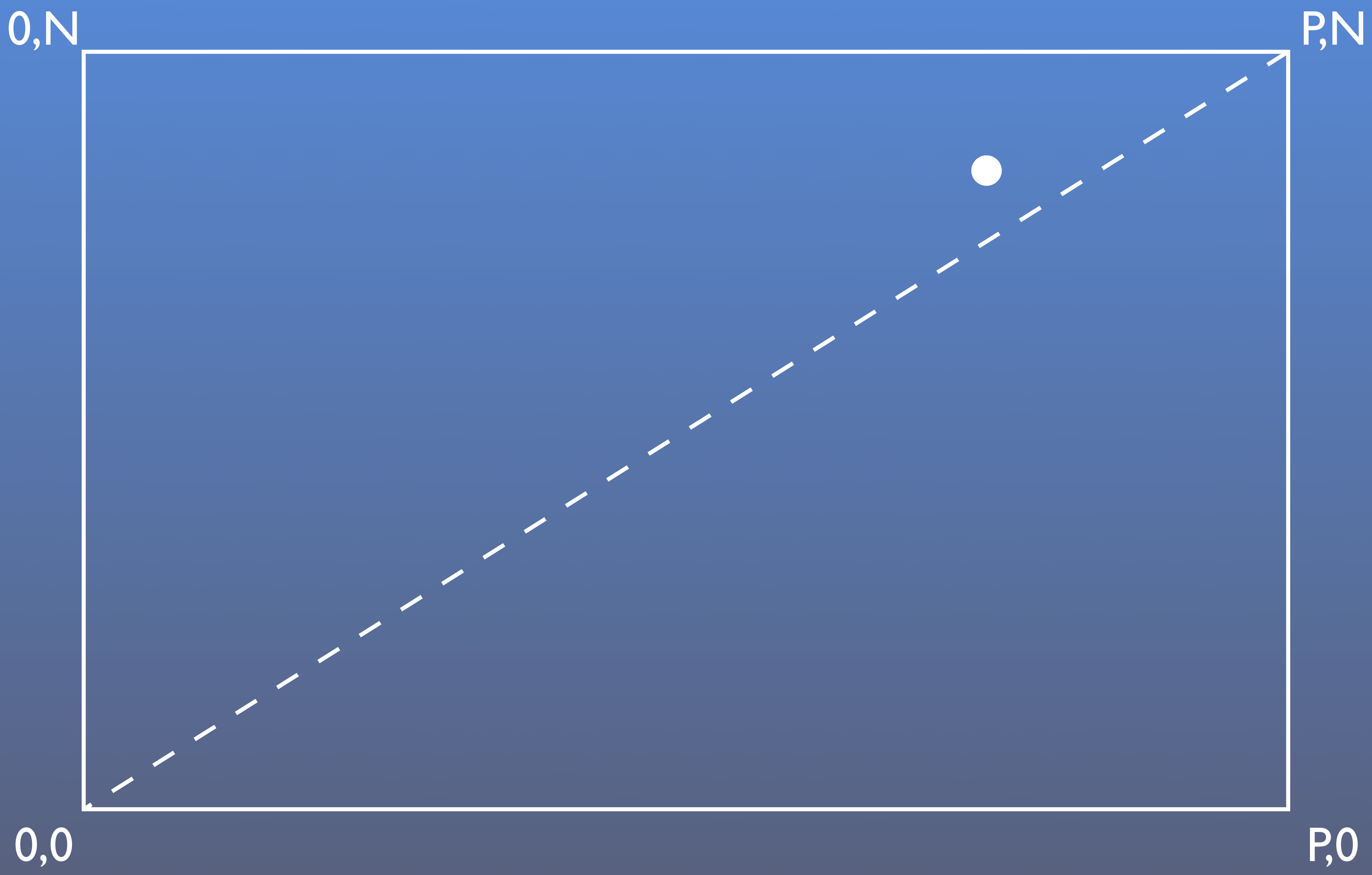
- There’s more
- Not limited to classification
- DTs manipulate data

# Manipulating Data (DTs)



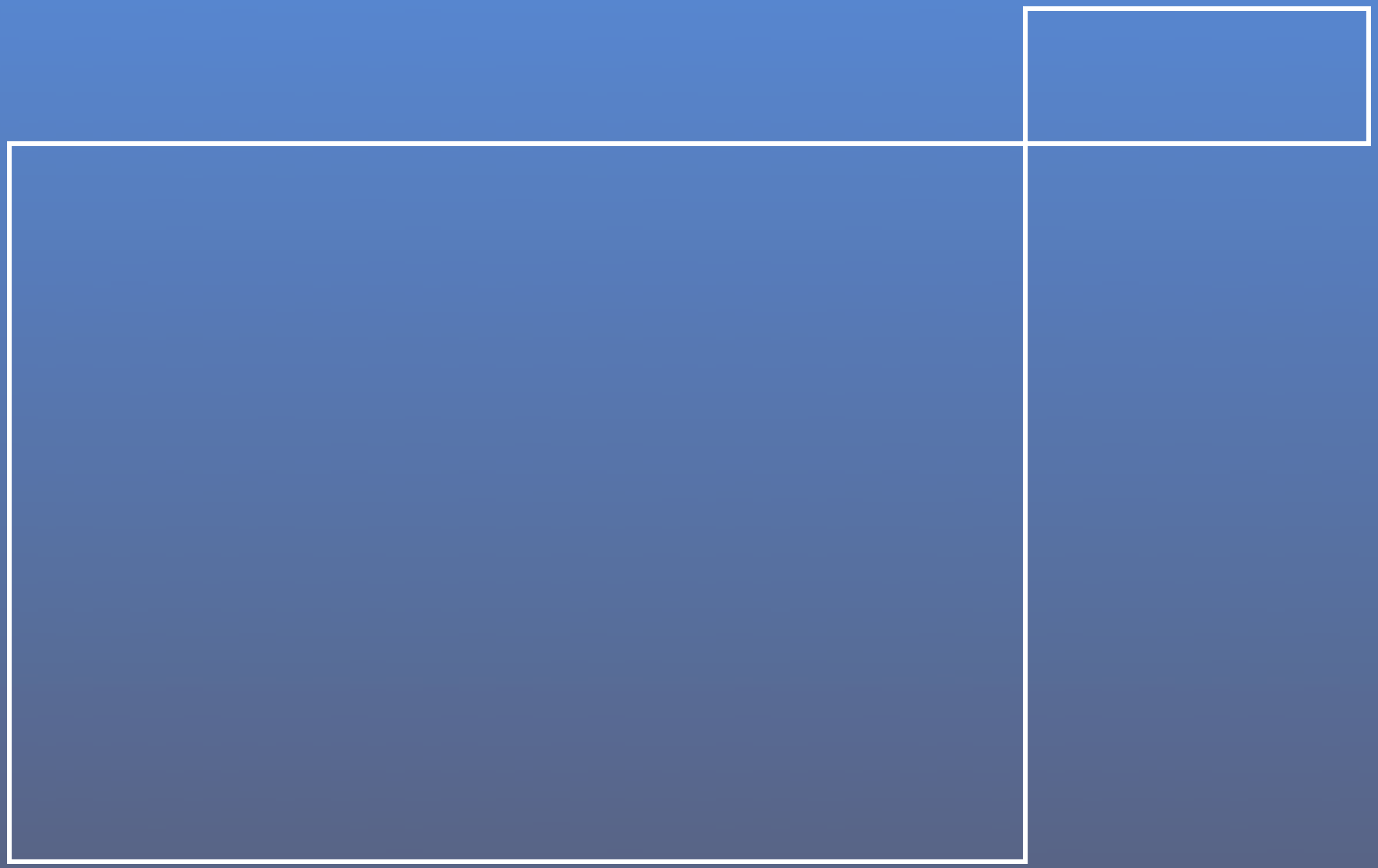
# Manipulating Data (DTs)

$$A_1 = v$$



# Manipulating Data (DTs)

$$A_1 = v$$



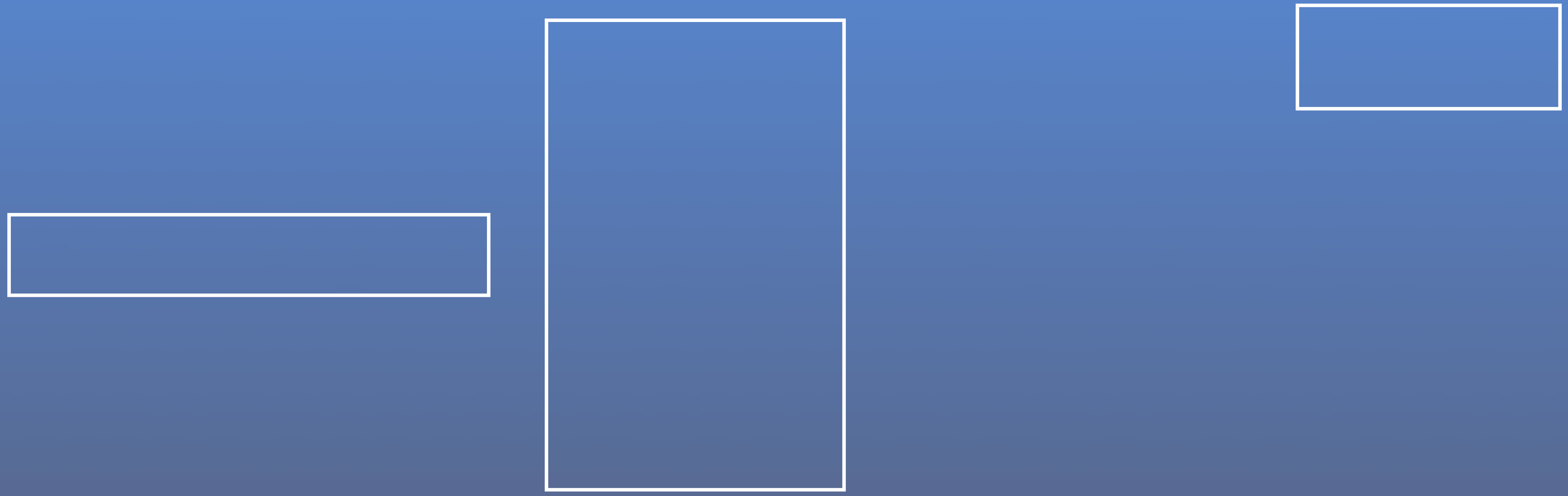
# Manipulating Data (DTs)

$$A_1 = v$$

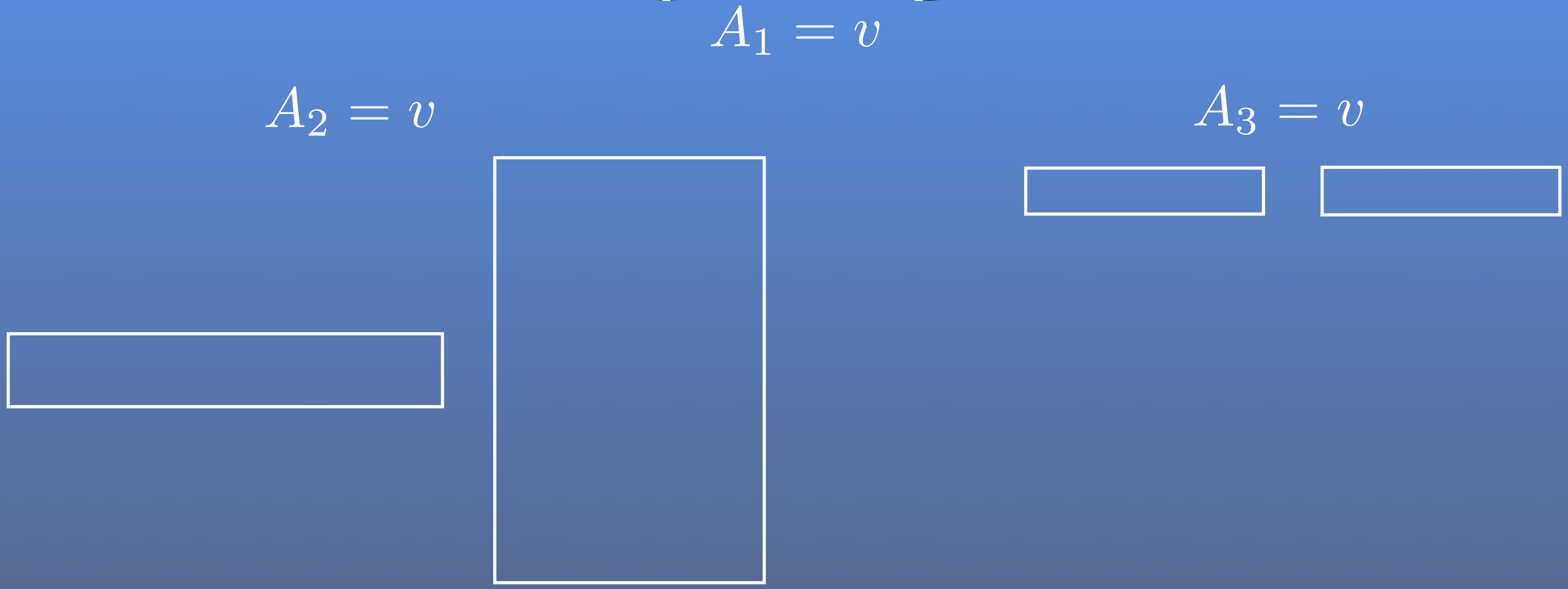
# Manipulating Data (DTs)

$A_1 = v$

$A_2 = v$



# Manipulating Data (DTs)

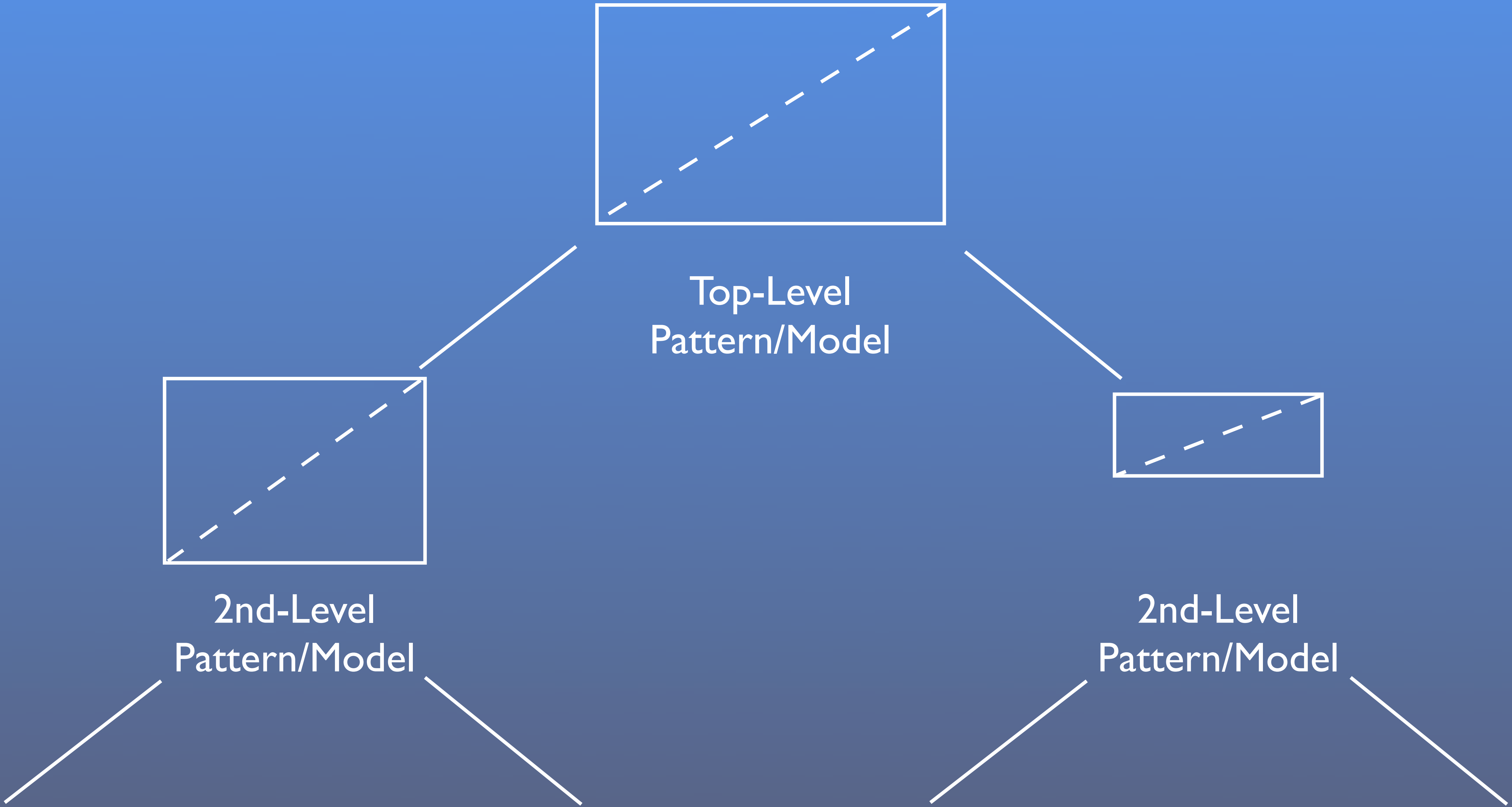


$A_1 = v$

$A_2 = v$

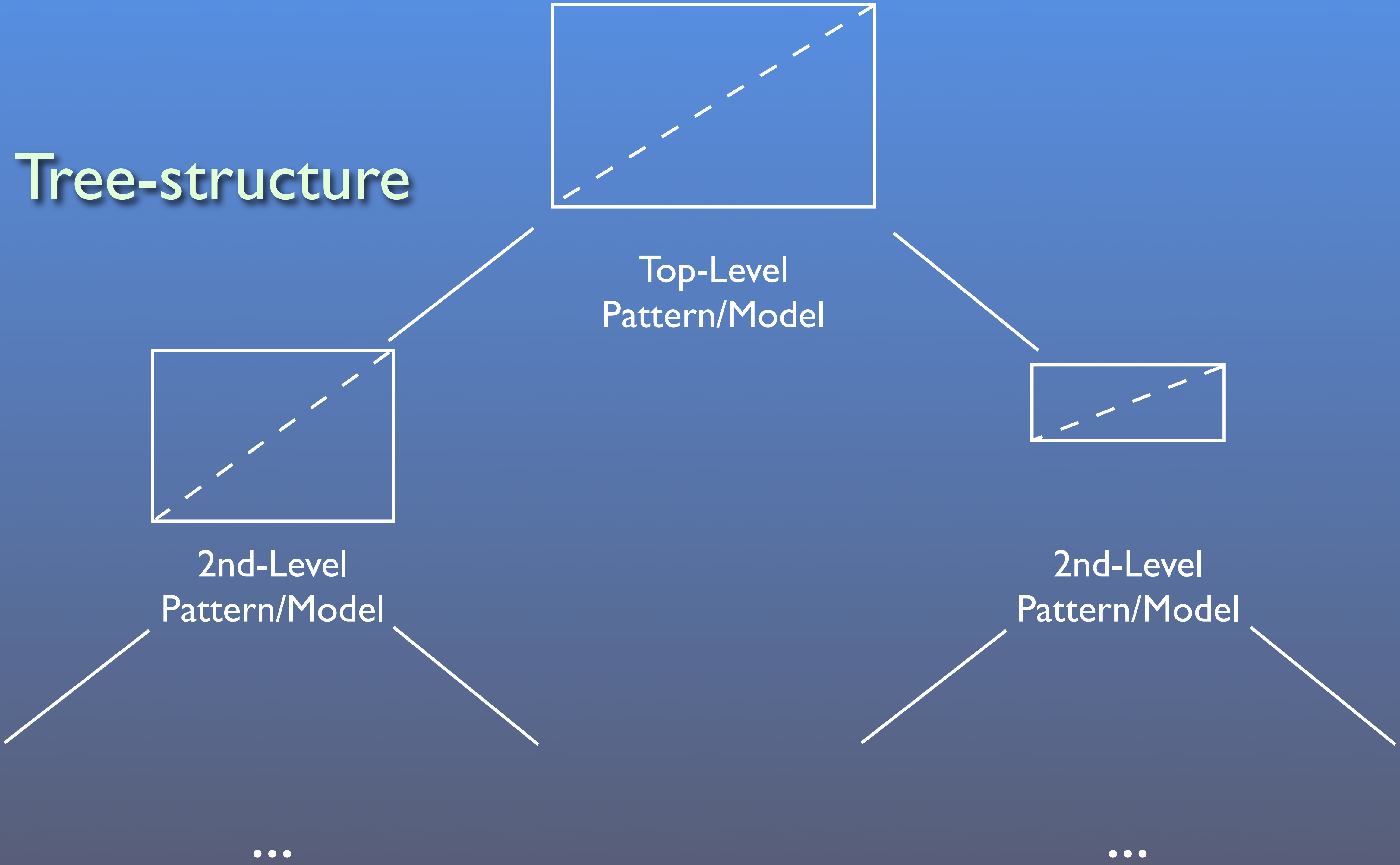
$A_3 = v$

# Tree-Ensembles



# Tree-Ensembles

Tree-structure



Top-Level Pattern/Model

2nd-Level Pattern/Model

2nd-Level Pattern/Model

…

…

# Exist already

- Not flat ensembles (sets)
- Difference in granularity/refinement
- Examples:

Mining interesting, non-redundant patterns:

Tree$^2$, DT-GBI

Building clustering trees (Dendrograms):

CobWeb, TIC, CG-Clus

Using ensembles of conjunctive rules for classification:

Ensemble Trees

# Conclusion

- Goal: improve accuracy, stability of DTs
- Invert ensemble process ⇒ inside nodes
- Simple to induce, effective pre-pruning
- Improvements necessary

# Conclusion

- Goal: improve accuracy, stability of DTs
- Invert ensemble process ⇒ inside nodes
- Simple to induce, effective pre-pruning
- Improvements necessary
- Instantiation of more general mechanism

Rebuttals?

Support?

Questions?

Doubt?