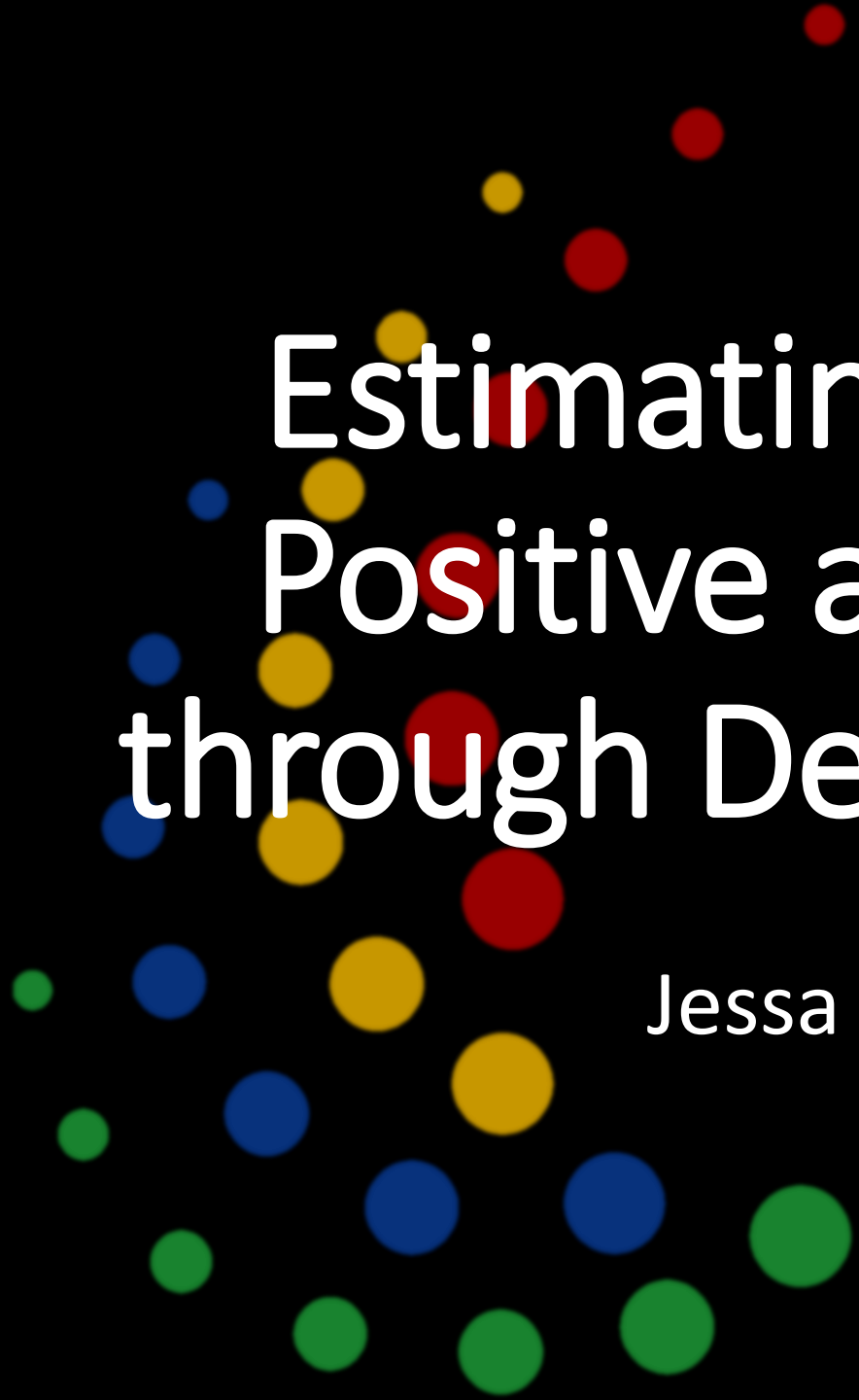


KU LEUVEN

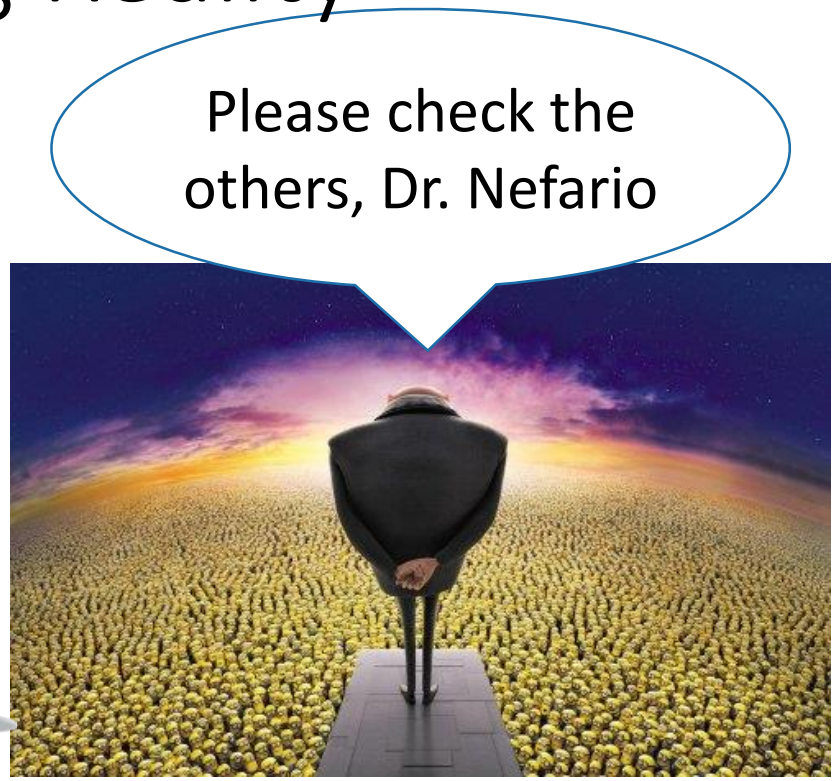
Estimating the Class Prior in Positive and Unlabeled Data through Decision Tree Induction

Jessa Bekker Jesse Davis

jessa.bekker@cs.kuleuven.be

people.cs.kuleuven.be/~jessa.bekker

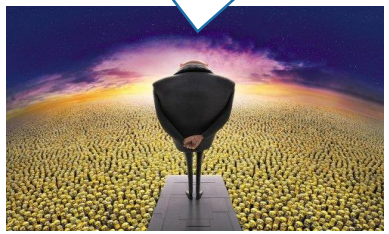
Machine Learning Reality



Learning with Positive and Unlabeled Data

These minions have diabetes

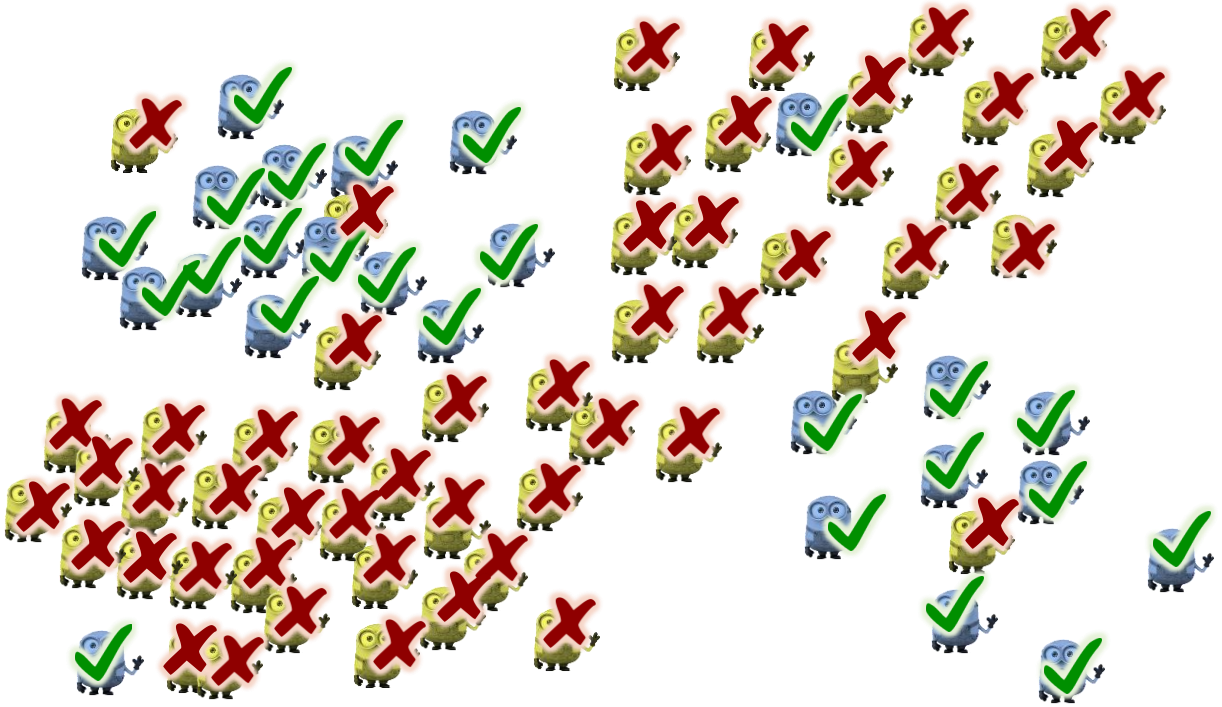
Please check the others, Dr. Nefario



Or….

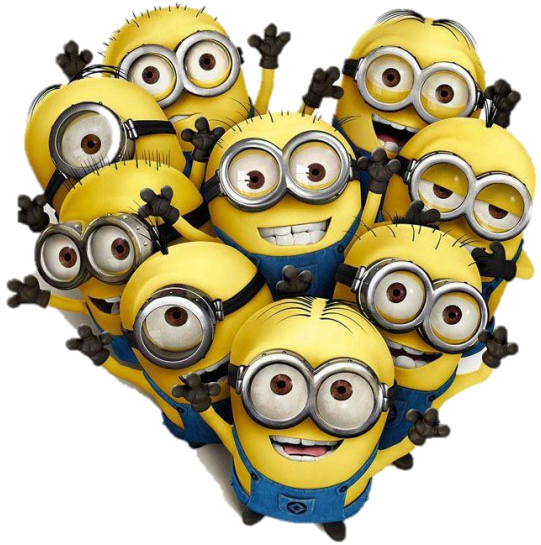
We can use the data as is, keeping in mind that the undiagnosed minions might still have diabetes.

Supervised

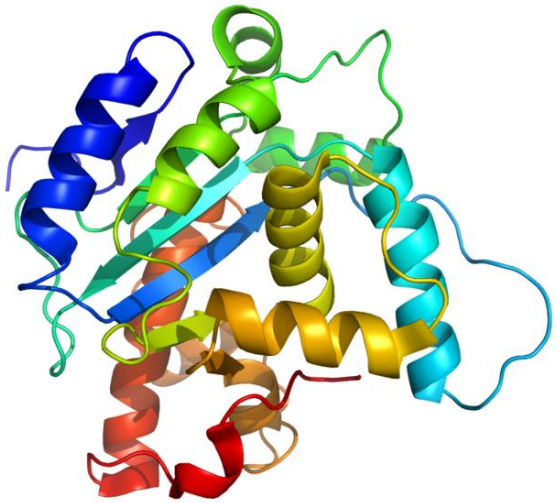


Positive and Unlabeled (PU)

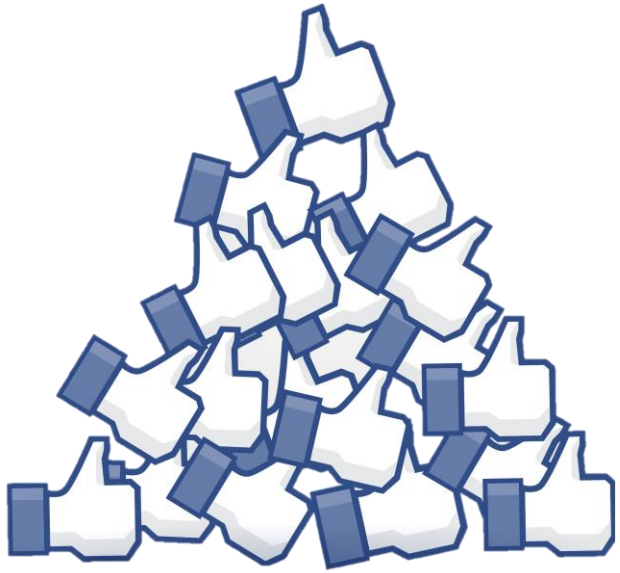
Positive and Unlabeled Data is Everywhere



Medical records



Incomplete gene/protein databases



Bookmarks/likes

Our Contribution

A new method for estimating the class prior α in PU data

$$\alpha = \Pr(positive)$$

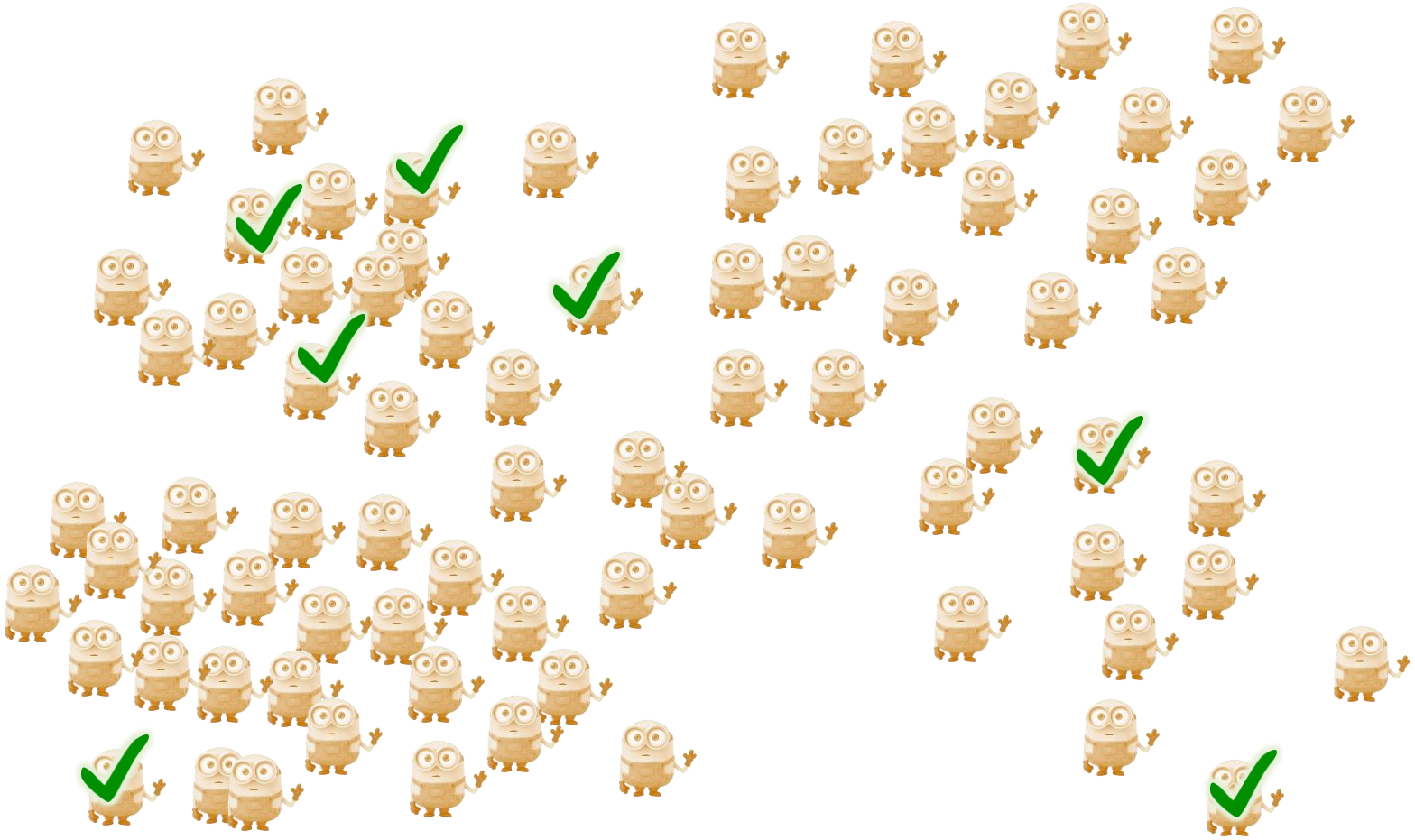
Why is this Important?

Knowing the class prior simplifies PU learning:
Standard learner modified with α = PU learner

Assumption: observed positive examples are *selected completely at random* from the positive set.

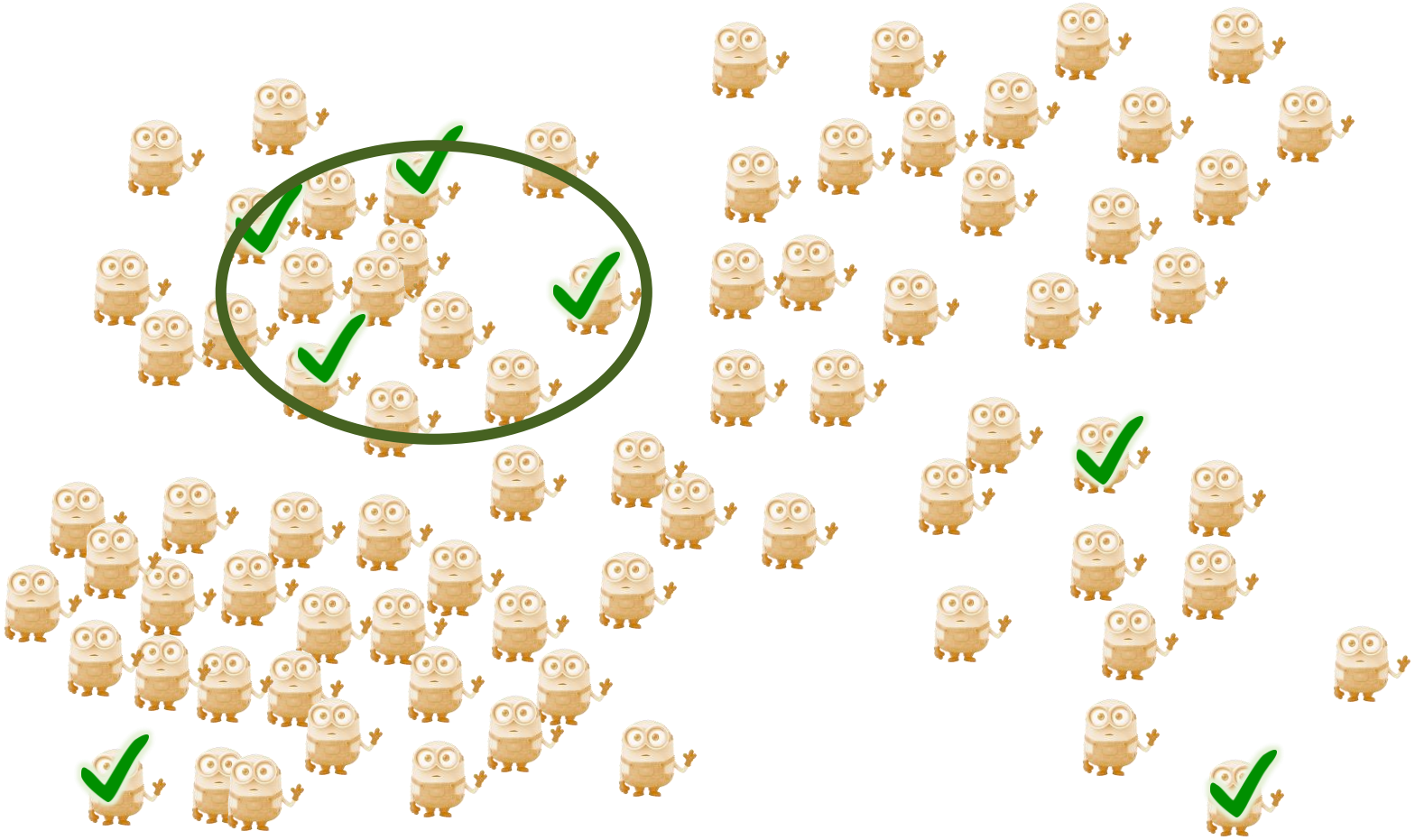
[Elkan and Noto, 2008]

Knowing the Class Prior Simplifies PU Learning



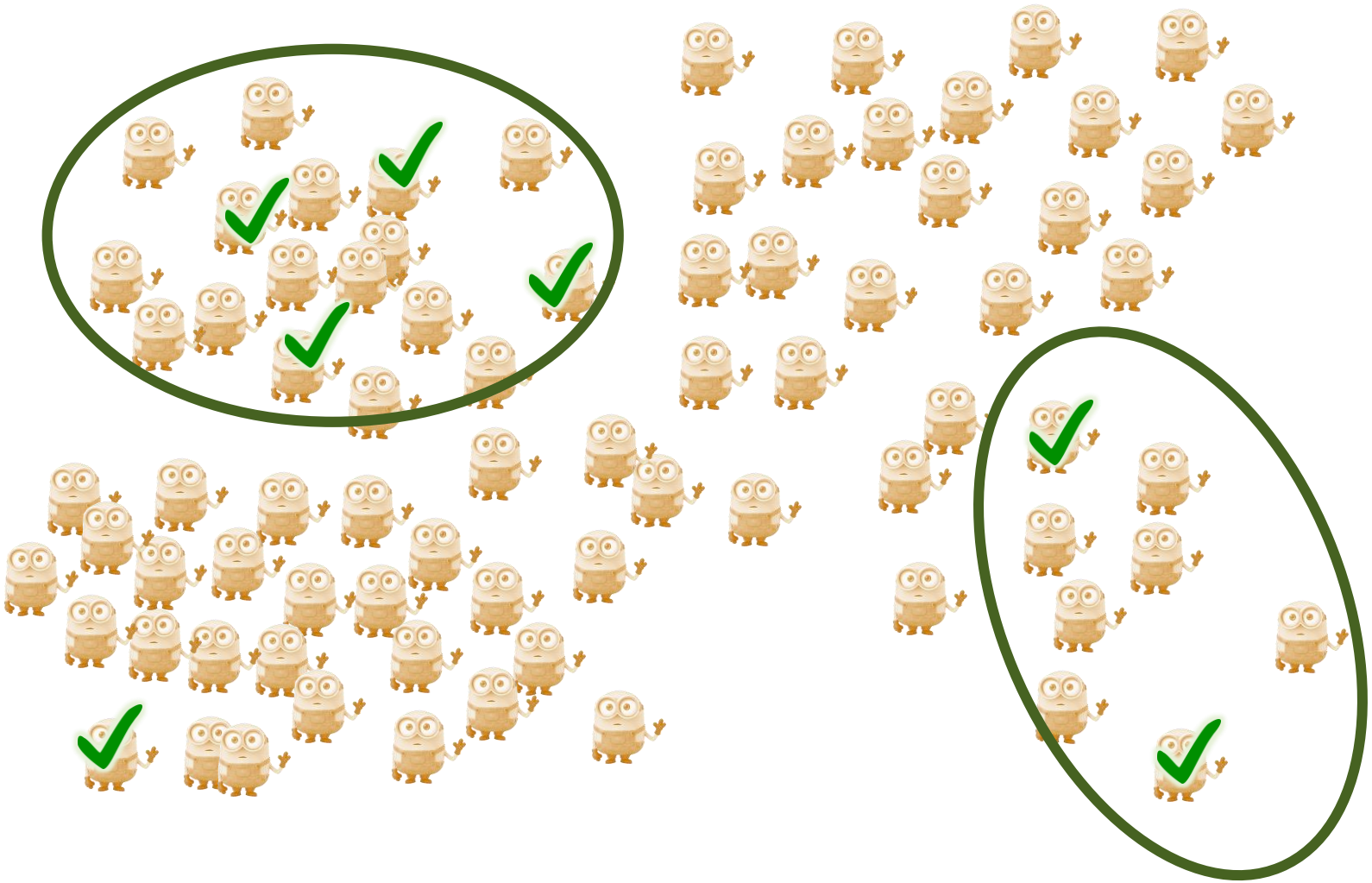
Knowing the Class Prior Simplifies PU Learning

$\alpha = 0.15$



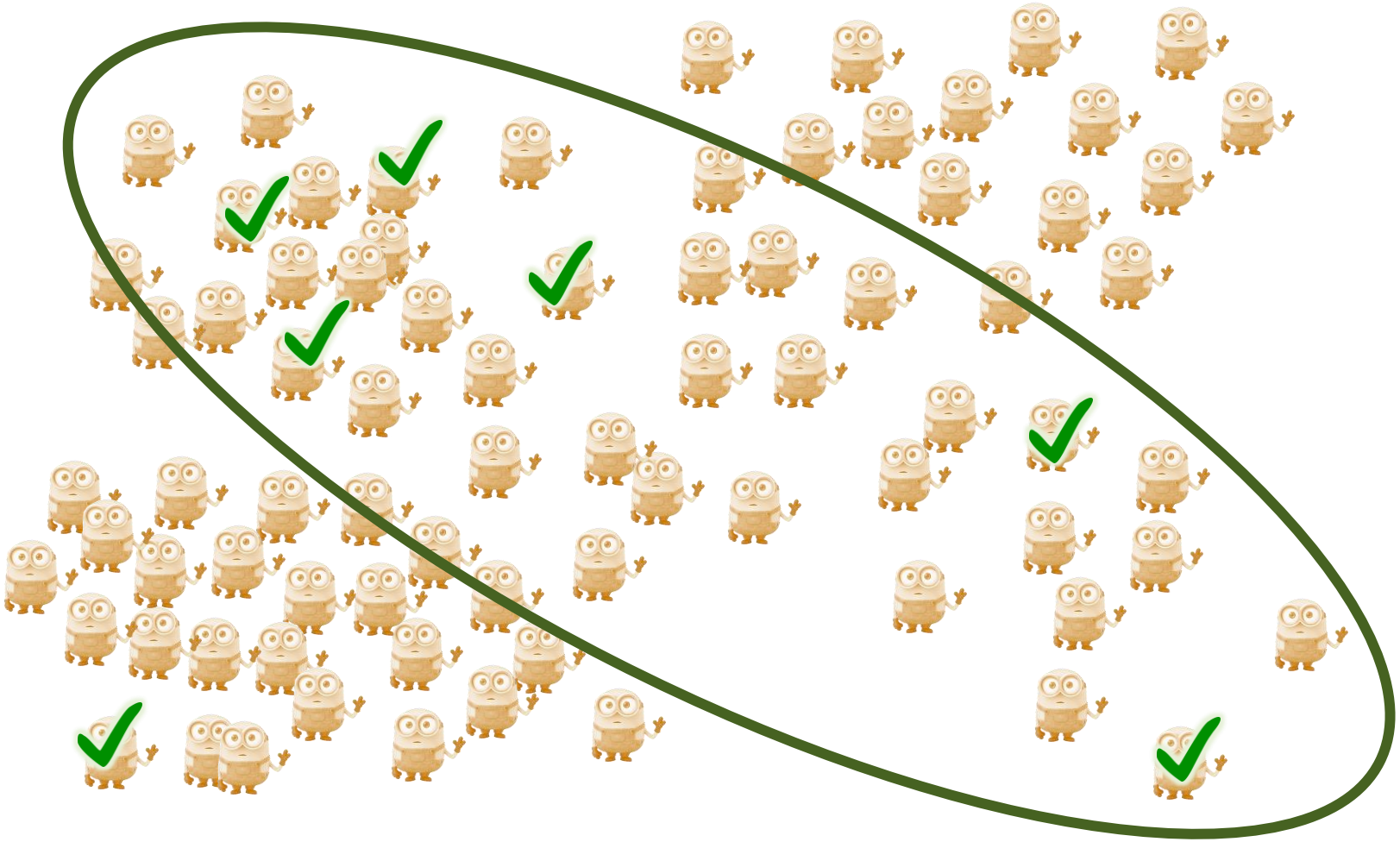
Knowing the Class Prior Simplifies PU Learning

$\alpha = 0.3$



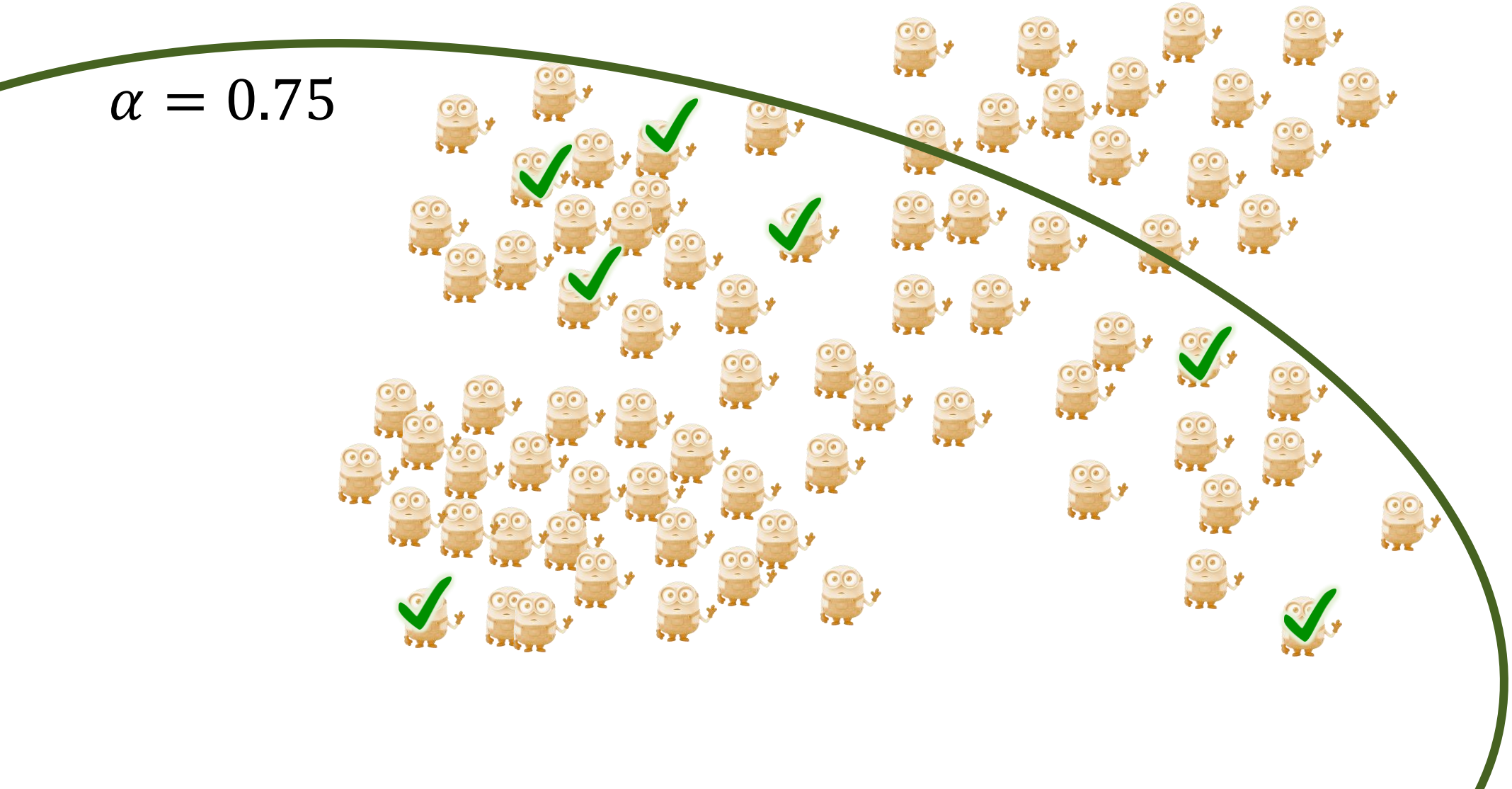
Knowing the Class Prior Simplifies PU Learning

$\alpha = 0.5$



Knowing the Class Prior Simplifies PU Learning

$\alpha = 0.75$



We Estimate the Label Frequency c

$$c = \Pr(labeled|positive)$$

Equivalent to estimating class prior α

$$c = \frac{\Pr(labeled, positive)}{\Pr(positive)} = \frac{\Pr(labeled)}{\Pr(positive)}$$

Count in data

Class prior α

Selected Completely at Random Assumption:

$$c = \Pr(labeled|positive) = \Pr(labeled|positive, x)$$

Our Method: TIcE

Tree Induction for *c* Estimation

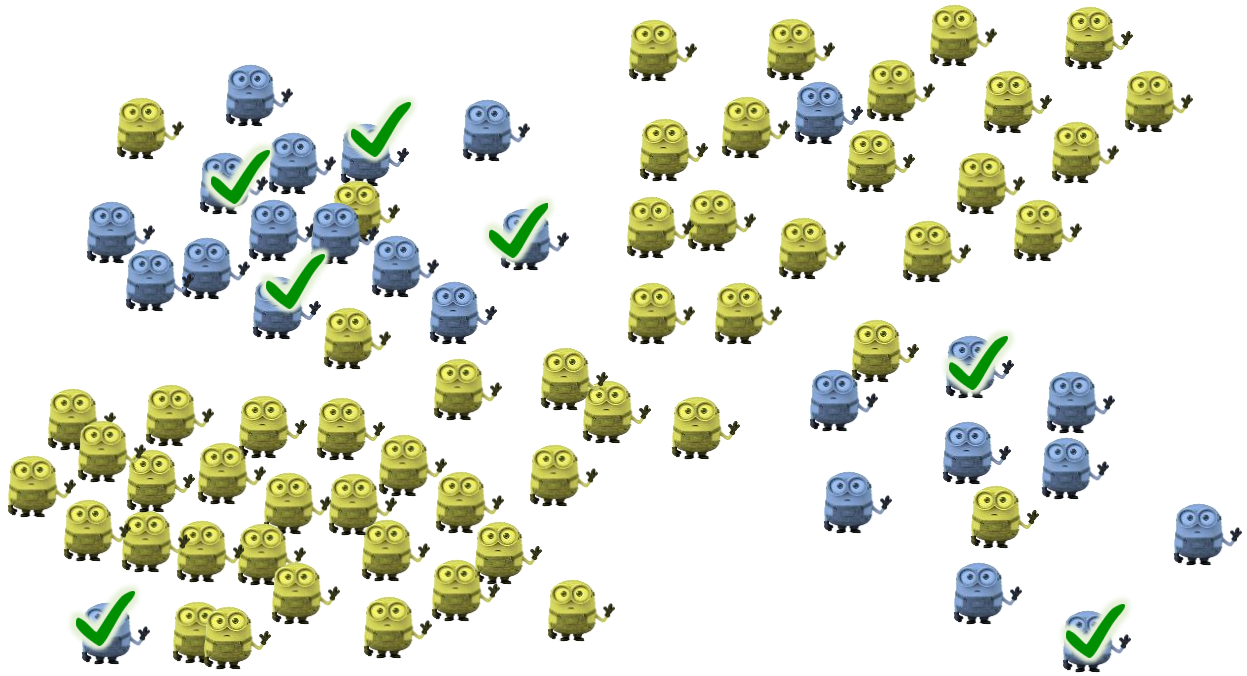
Insight 1: Data Implies Lower Bound on c

Upper bound on #positives

$$\#positive \leq \#total$$

➡ Lower bound on c

$$c = \frac{\#labeled}{\#positive} \geq \frac{\#labeled}{\#total}$$



Insight 1: Data Implies Lower Bound on c

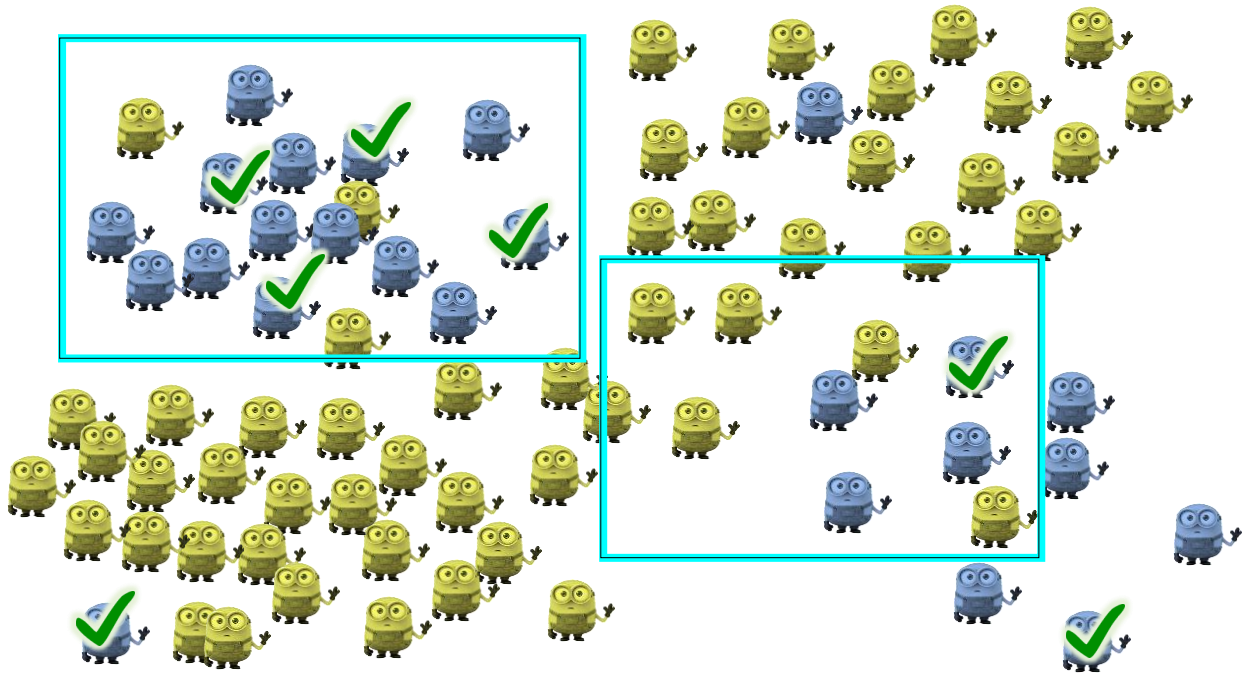
Upper bound on #positives

$$\#positive_{\boldsymbol{i}} \leq \#total_{\boldsymbol{i}}$$

➡ Lower bound on c

Same c in any subset i!

$$c = \frac{\#labeled_{\boldsymbol{i}}}{\#positive_{\boldsymbol{i}}} \geq \frac{\#labeled_{\boldsymbol{i}}}{\#total_{\boldsymbol{i}}}$$



Insight 1: Data Implies Lower Bound on c

Upper bound on #positives

$$\#positive_i \leq \#total_i$$

➡ Lower bound on c

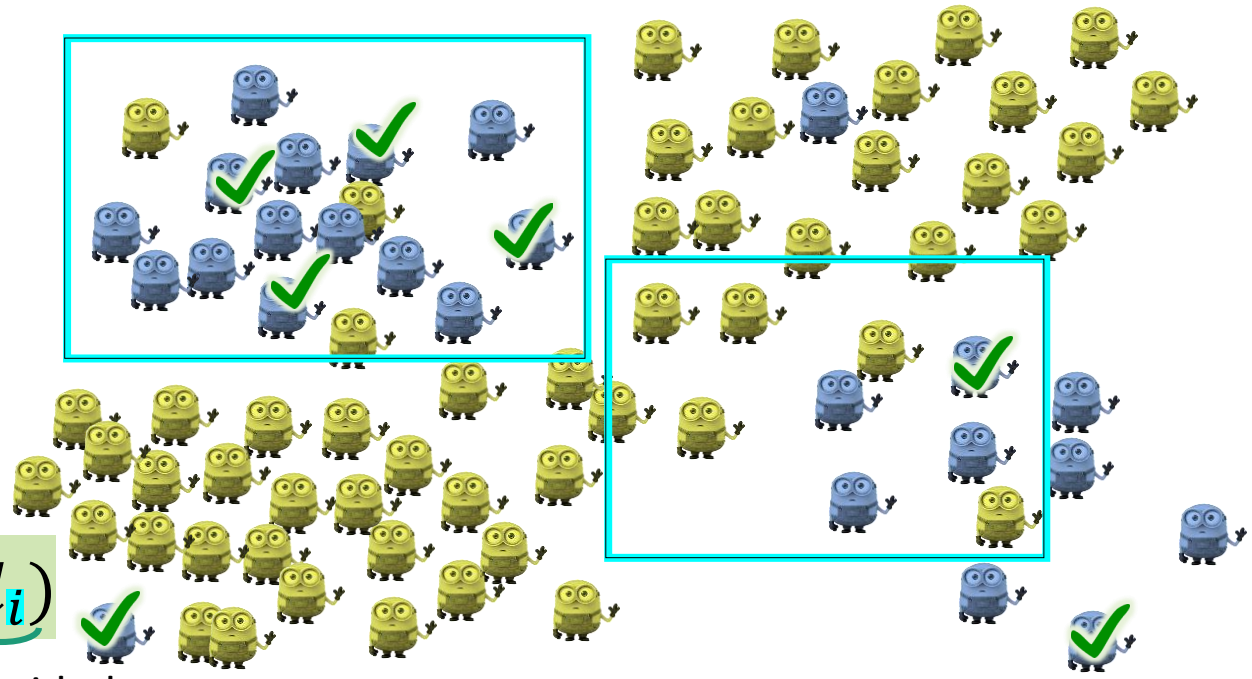
Same c in any subset i!

Empirically, c varies

$$c = \frac{\#labeled_i}{\#positive_i}$$

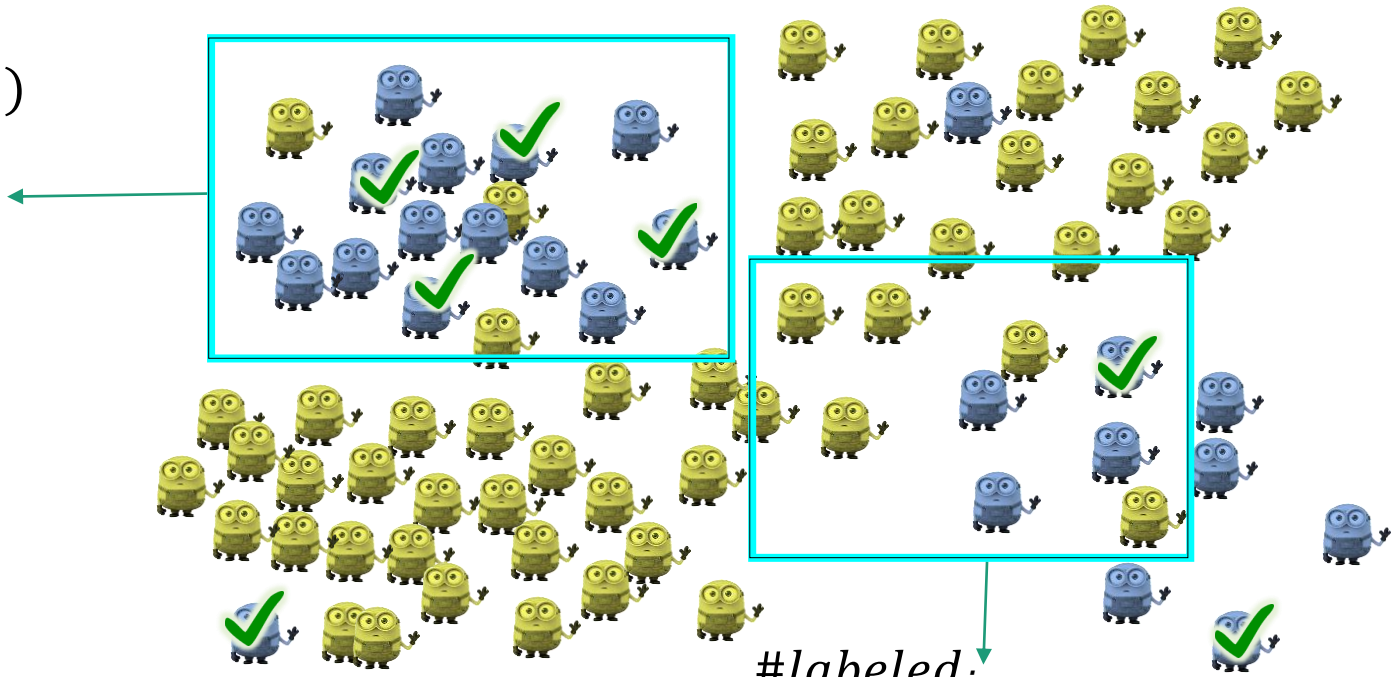
$$\geq \frac{\#labeled_i}{\#total_i} \underbrace{-\varepsilon(\#total_i)}$$

Error term from 1-sided Chebyshev inequality



Insight 2: Positive Subsets Give Very Tight Bounds

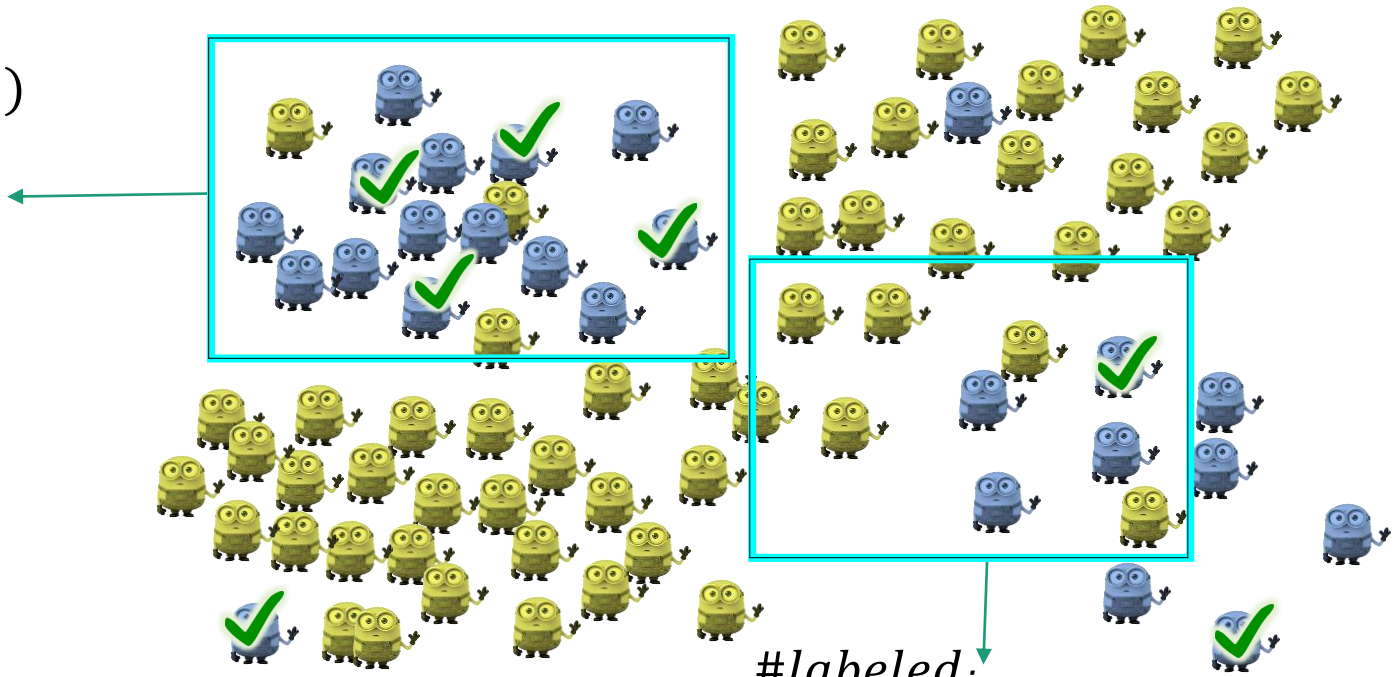
$$c \geq \frac{\#labeled_i}{\#total_i} - \varepsilon(\#total_i)$$
$$\geq 0.24 \; - \varepsilon(17)$$



$$c \geq \frac{\#labeled_i}{\#total_i} - \varepsilon(\#total_i)$$
$$\geq 0.11 \; - \varepsilon(9)$$

Insight 3: Highly Labeled Subsets are Likely Positive

$$c \geq \frac{\#labeled_i}{\#total_i} - \varepsilon(\#total_i)$$
$$\geq 0.24 \; - \varepsilon(17)$$



$$c \geq \frac{\#labeled_i}{\#total_i} - \varepsilon(\#total_i)$$
$$\geq 0.11 \; - \varepsilon(9)$$

Tree Induction for *c* Estimation (TI*c*E)

Step 1: Look for highly labeled subsets

using decision tree induction

Step 2: Estimate *c*

by taking the maximum lower bound of subsets

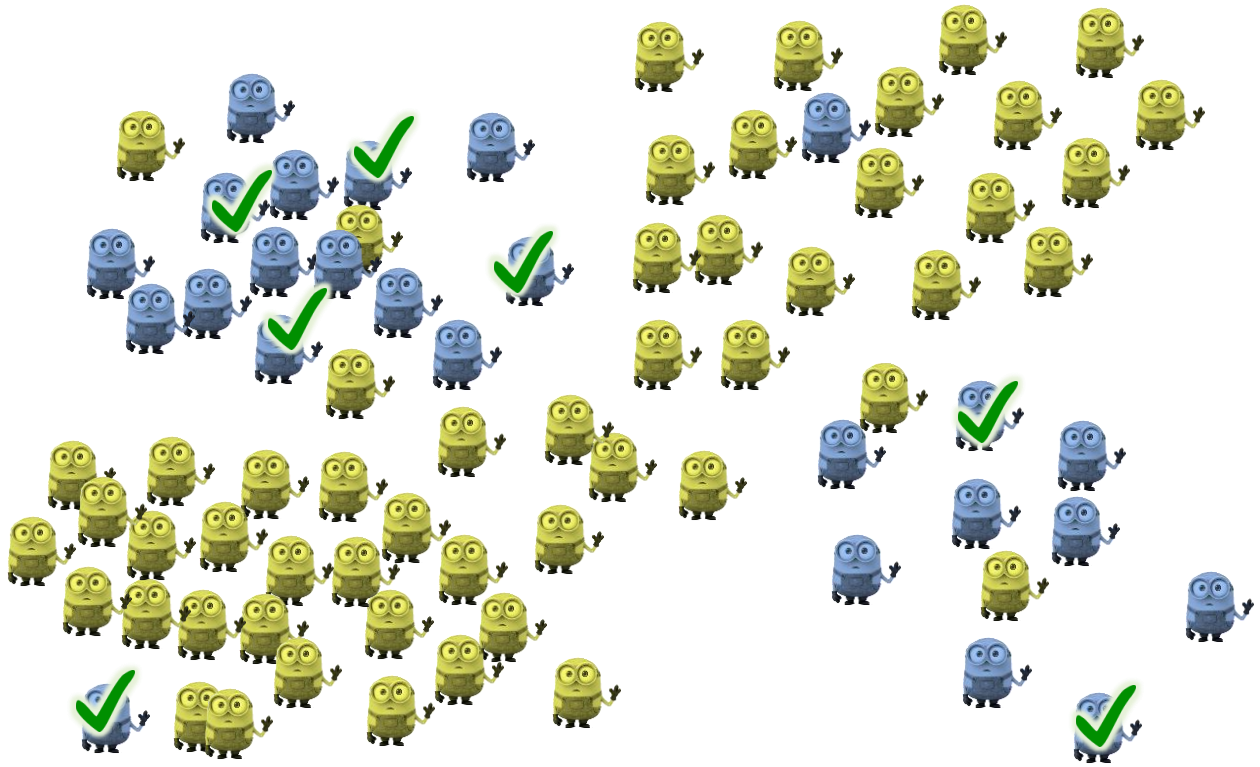
TIcE: Look for Highly Labeled Subsets

Split criterion: max-bepp

$$\max_i \frac{\#labeled_i}{\#\ total_i + k}$$

Subset with largest labeled proportion

Penalizes small subsets



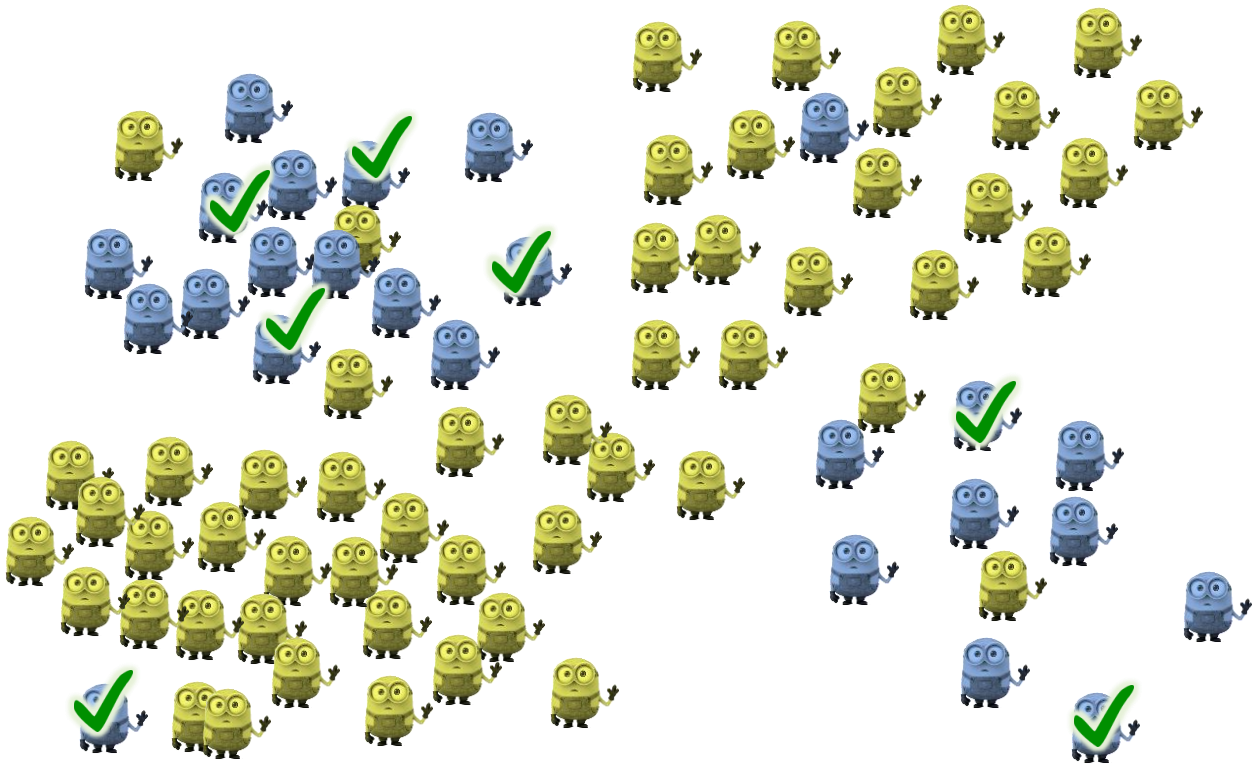
TIcE: Look for Highly Labeled Subsets

Split criterion: max-bepp

$$\max_i \frac{\#labeled_i}{\#\ total_i + k}$$

Subset with largest labeled proportion

Penalizes small subsets



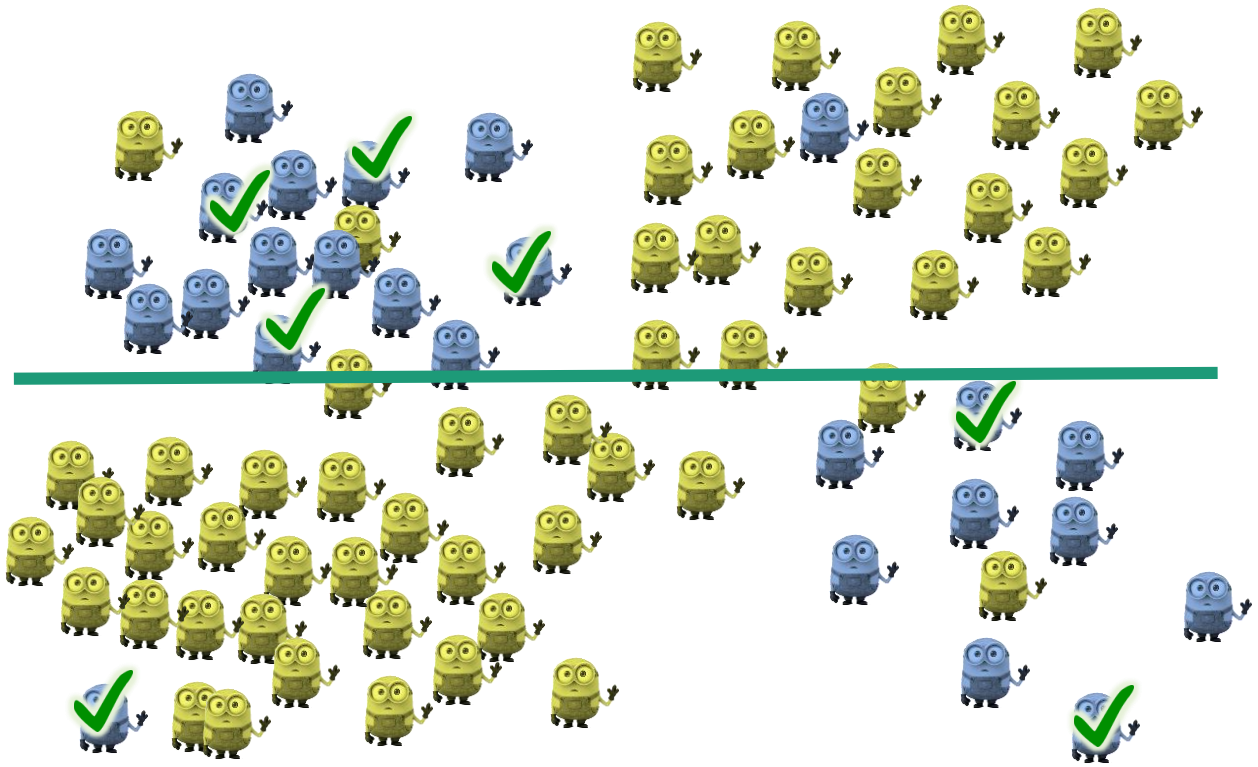
TIcE: Look for Highly Labeled Subsets

Split criterion: max-bepp

$$\max_i \frac{\#labeled_i}{\#\ total_i + k}$$

Subset with largest labeled proportion

Penalizes small subsets



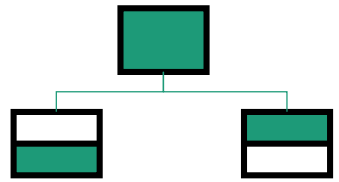
TIcE: Look for Highly Labeled Subsets

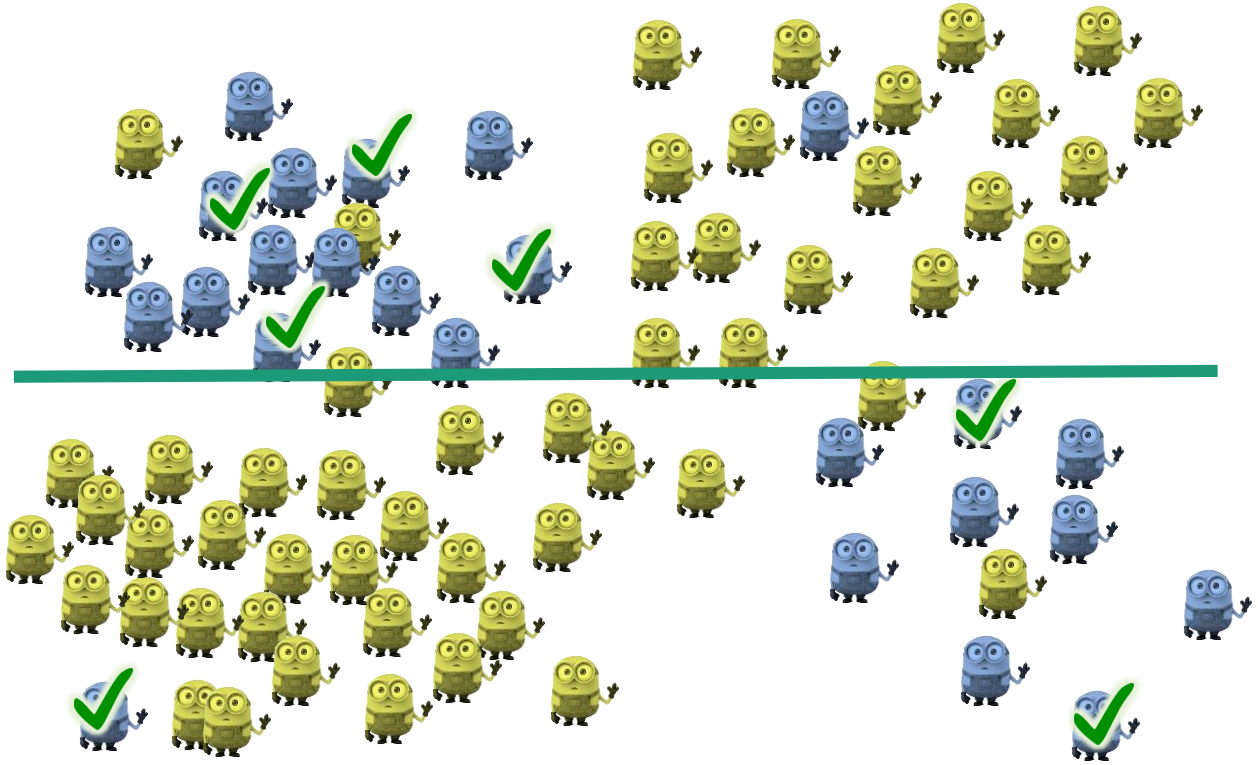
Split criterion: max-bepp

$$\max_i \frac{\#labeled_i}{\#\ total_i + k}$$

Subset with largest labeled proportion

Penalizes small subsets





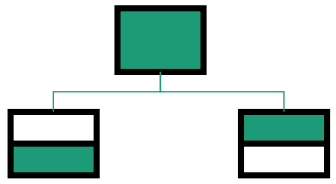
TIcE: Look for Highly Labeled Subsets

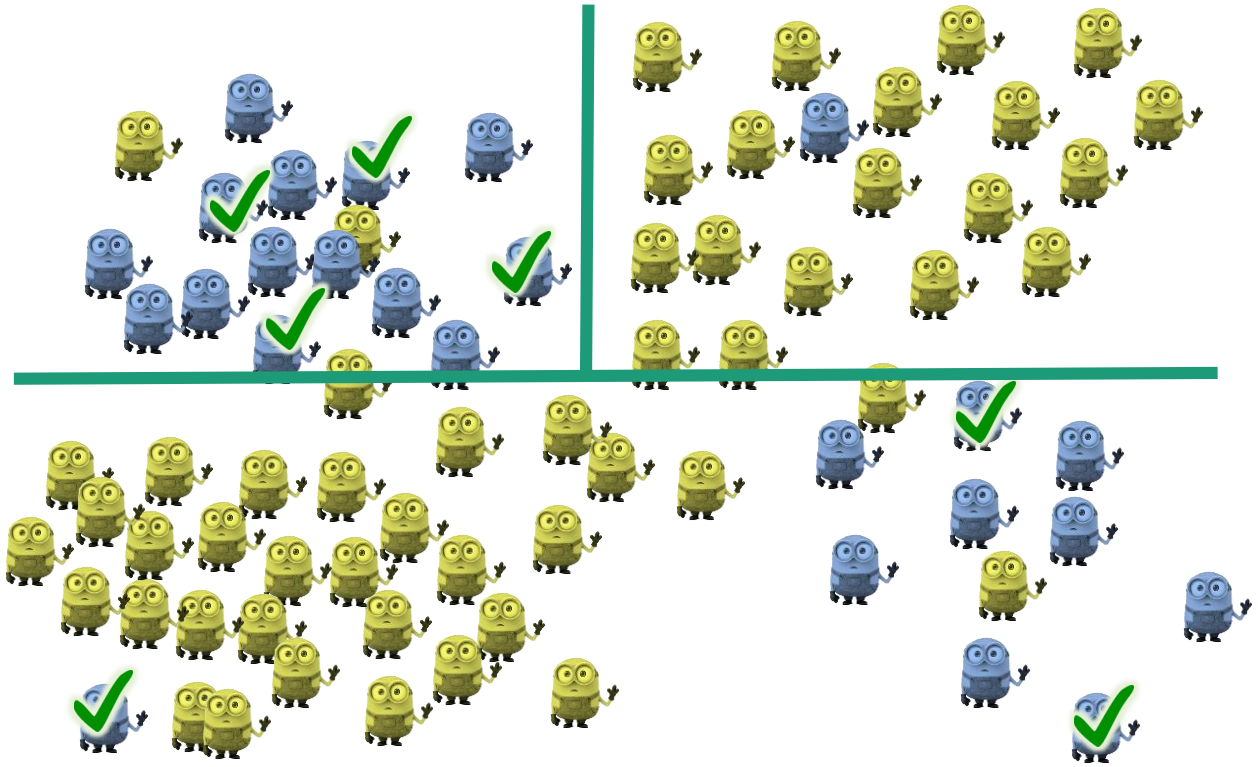
Split criterion: max-bepp

$$\max_i \frac{\#labeled_i}{\#\ total_i + k}$$

Subset with largest labeled proportion

Penalizes small subsets





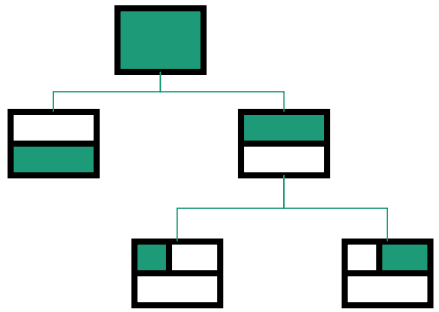
TIcE: Look for Highly Labeled Subsets

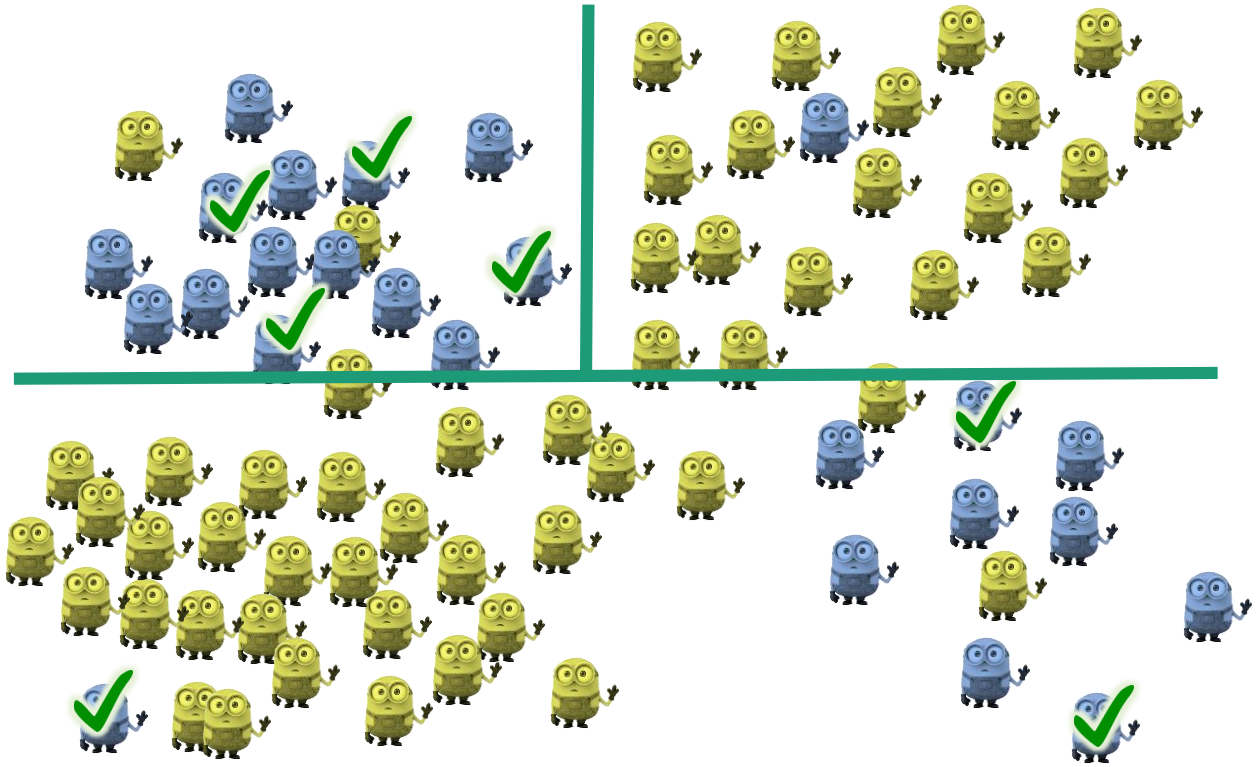
Split criterion: max-bepp

$$\max_i \frac{\#labeled_i}{\#\ total_i + k}$$

Subset with largest labeled proportion

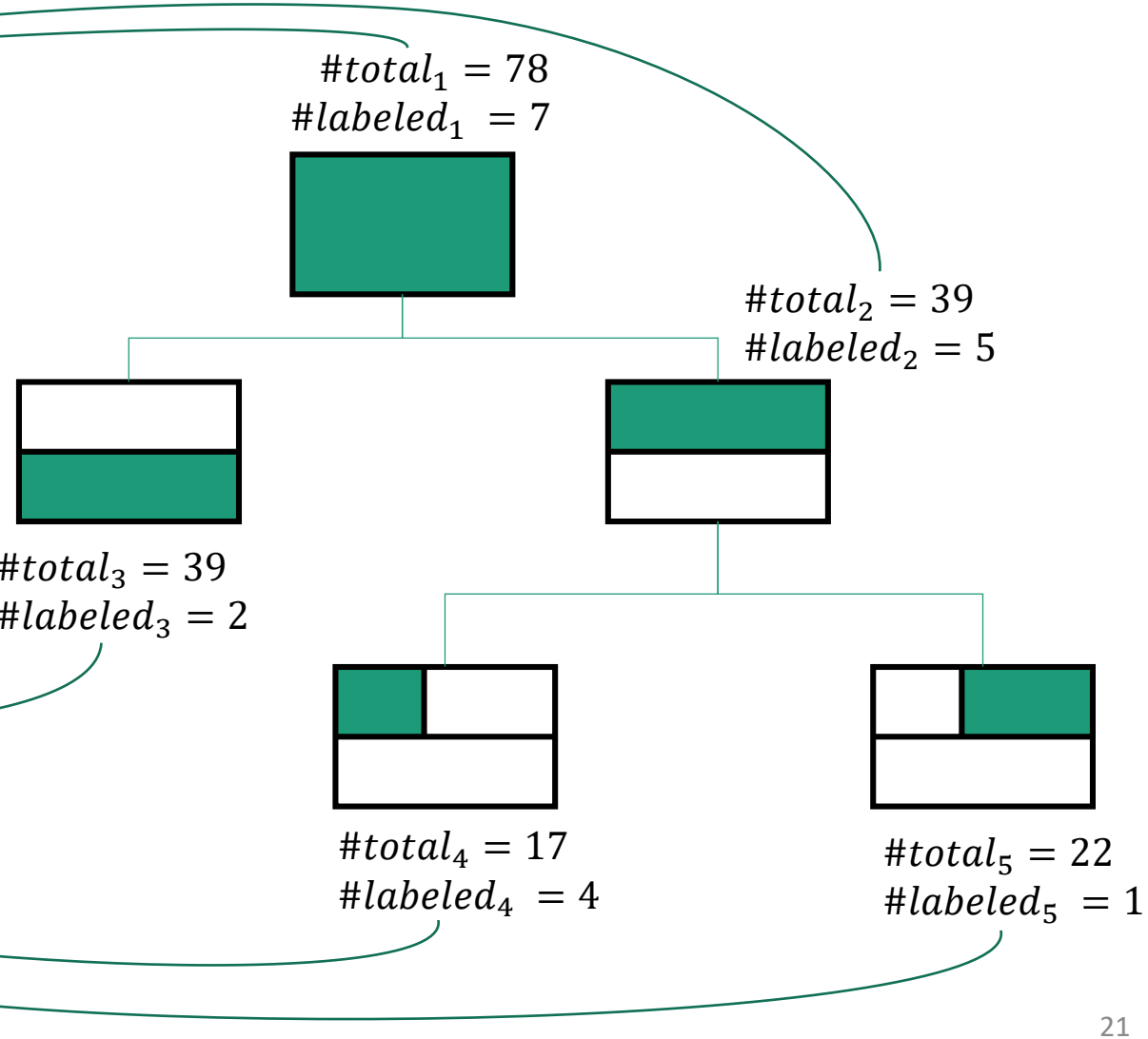
Penalizes small subsets





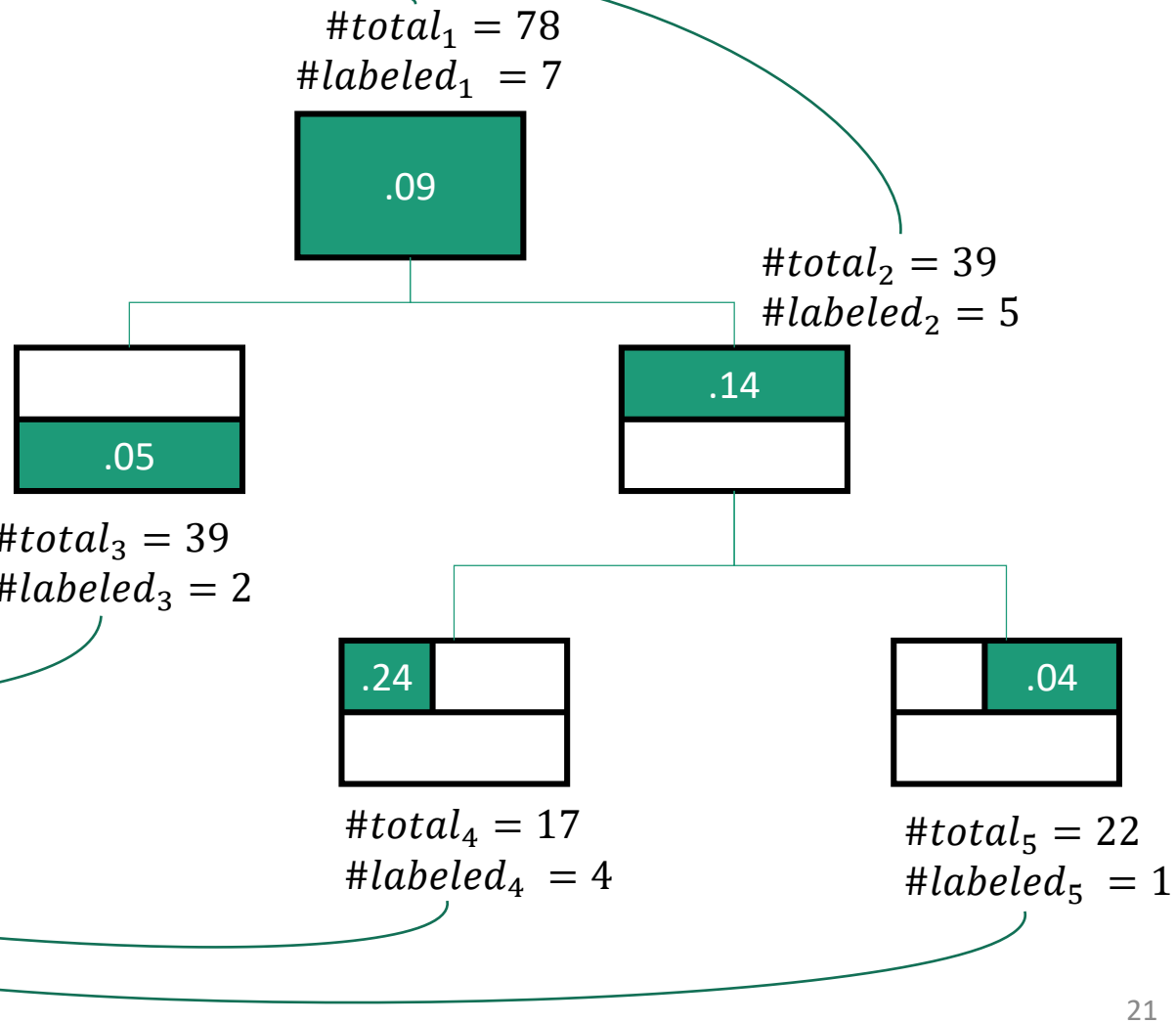
TIcE: c Estimation with Tree-Implied Subsets

$$c = \max_i \frac{\#labeled_i}{\#total_i} - \varepsilon(\#total_i)$$



TIcE: c Estimation with Tree-Implied Subsets

$$c = \max_i \frac{\#labeled_i}{\#total_i} - \varepsilon(\#total_i)$$



TIcE: Prevent Overfitting

Selecting subsets based on labels

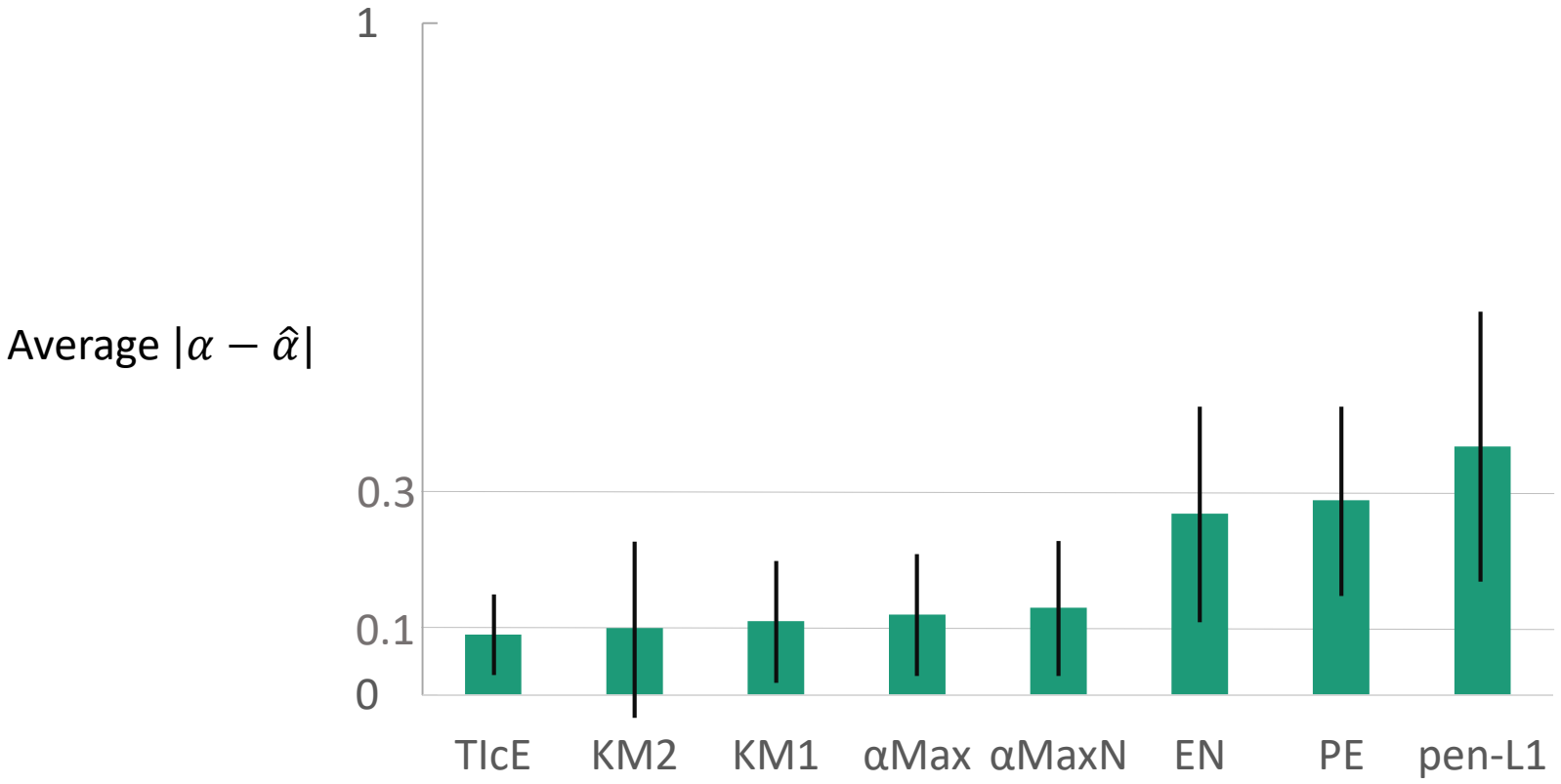
⇒ likely to find subsets with a higher empirical label frequency.

Solution:

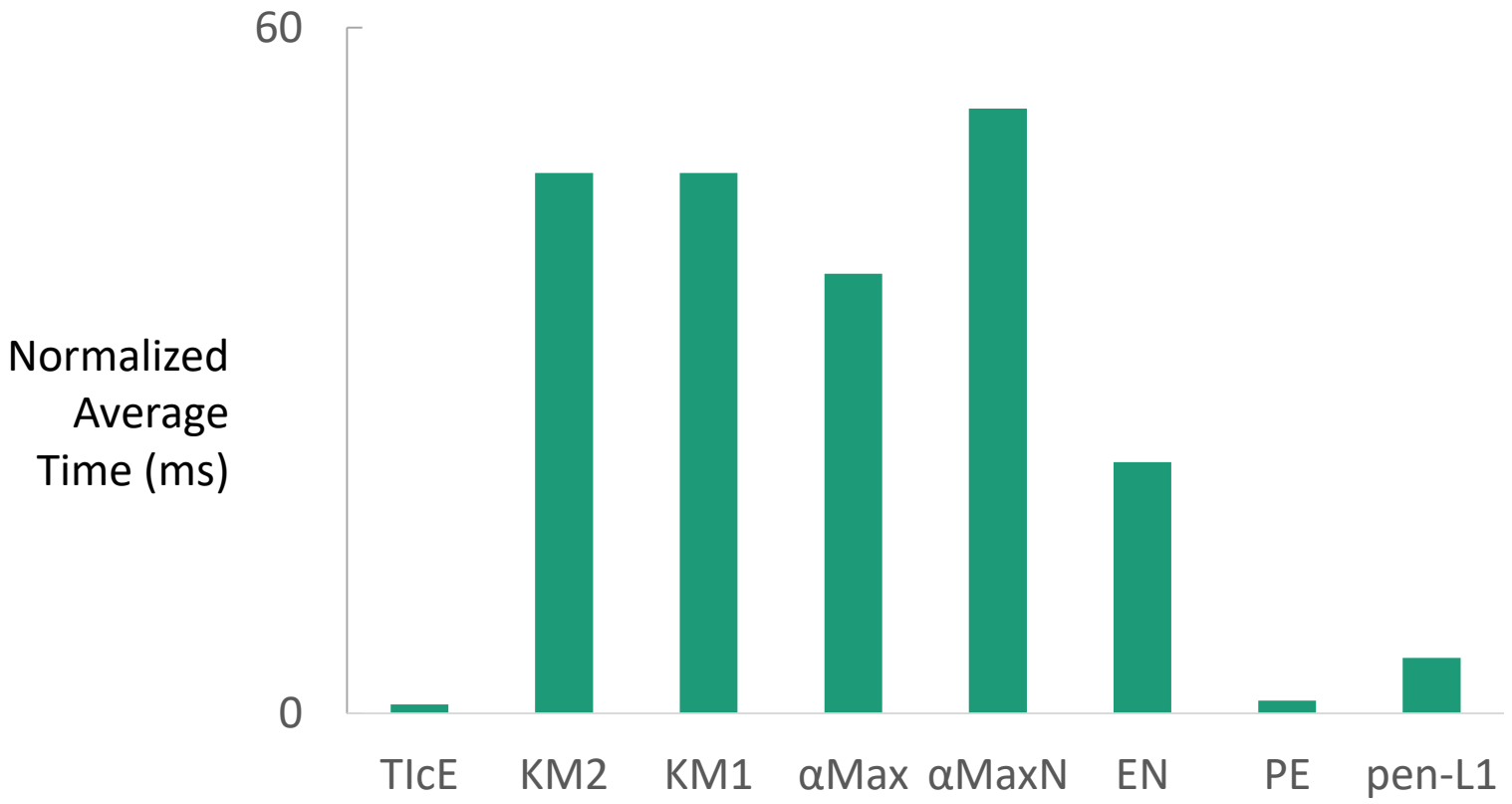
Different datasets for tree induction and c estimation

∼ k-fold cross validation

TIcE is Accurate



TIcE is Fast


60
Normalized Average Time (ms)
0
TIcE
KM2
KM1
αMax
αMaxN
EN
PE
pen-L1

Conclusions

- PU learning is very useful in practice
- Knowing the class prior α

  makes PU learning easier
- TI*c*E estimates α from PU data
  - Simple
  - Accurate
  - Fast