Class Prior Estimation in Active Positive and Unlabeled Learning

Lorenzo Perini, *Vincent Vercruyssen, Jesse Davis*

IJCAI-PRICAI 2020

https://people.cs.kuleuven.be/~lorenzo.perini/

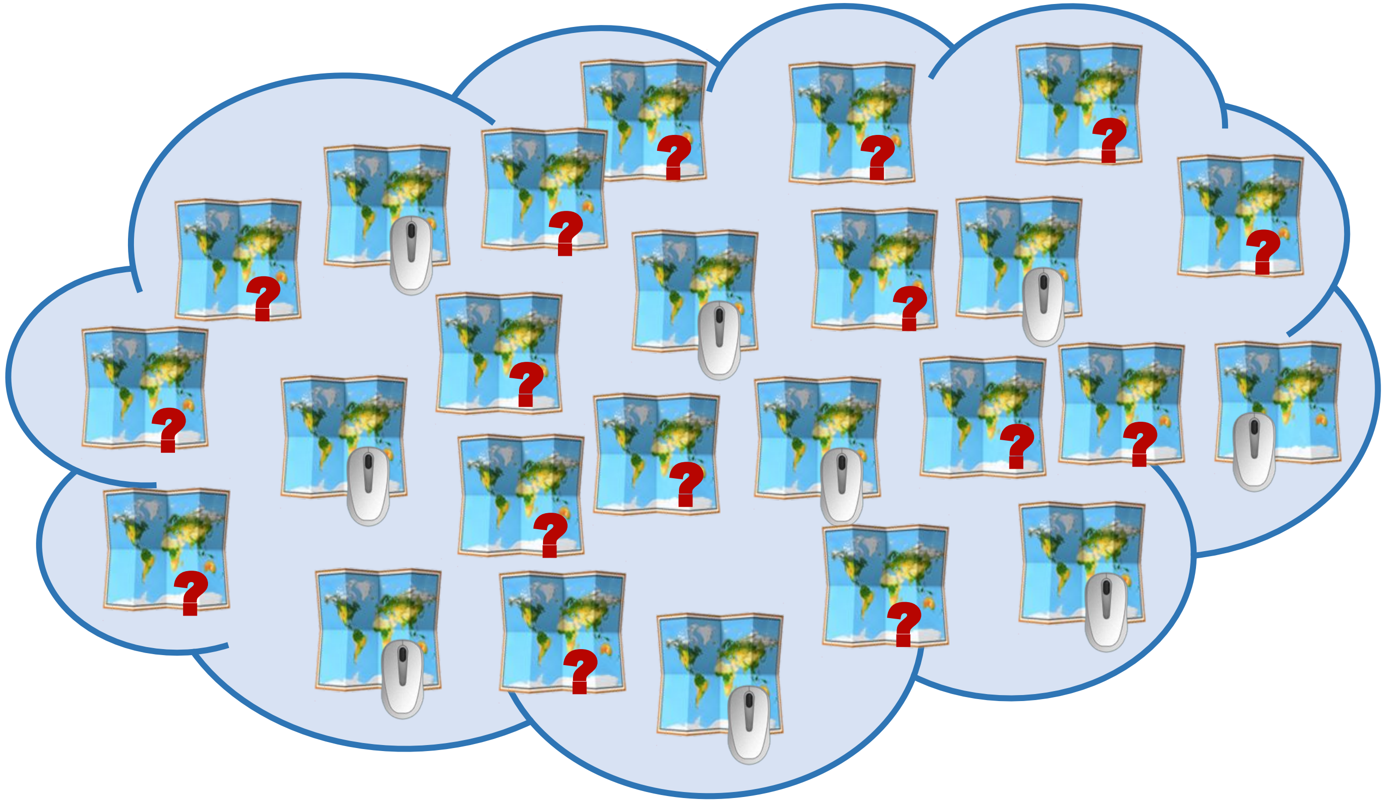
@LorenzoPerini95

Positive and Unlabeled (PU) Data Arises Naturally. For Example:

Task: *Estimate the relevance of ads based on click data*

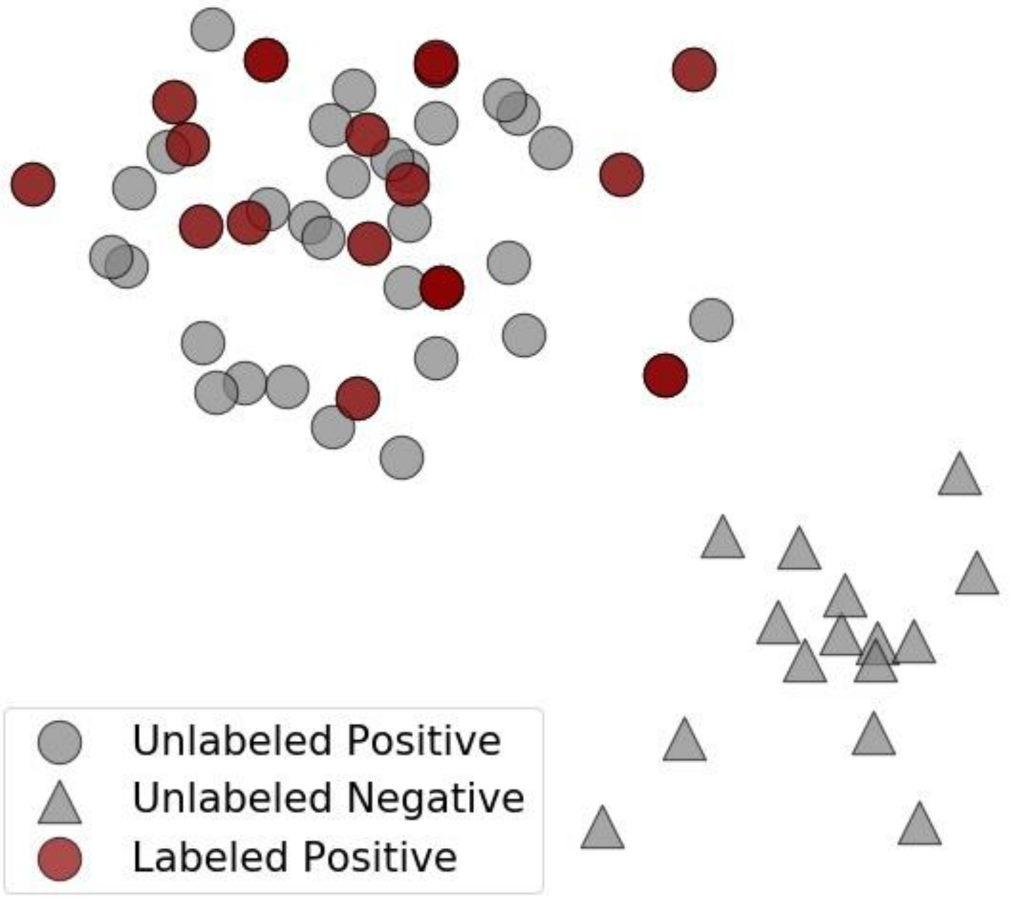
Challenge: *Ads that the user has not clicked on may still be relevant!*

Positive (click)
Unlabeled (no click)



How can we learn from PU data?

Common Approach to Learning from PU Data Revolves around Estimating the Class Prior



Most common approaches estimate the class prior by assuming labels are Selected Completely At Random (**SCAR**)

$$\mathbb{P}(labeled \,|\, positive,\, x) = \mathbb{P}(labeled \,|\, positive)$$

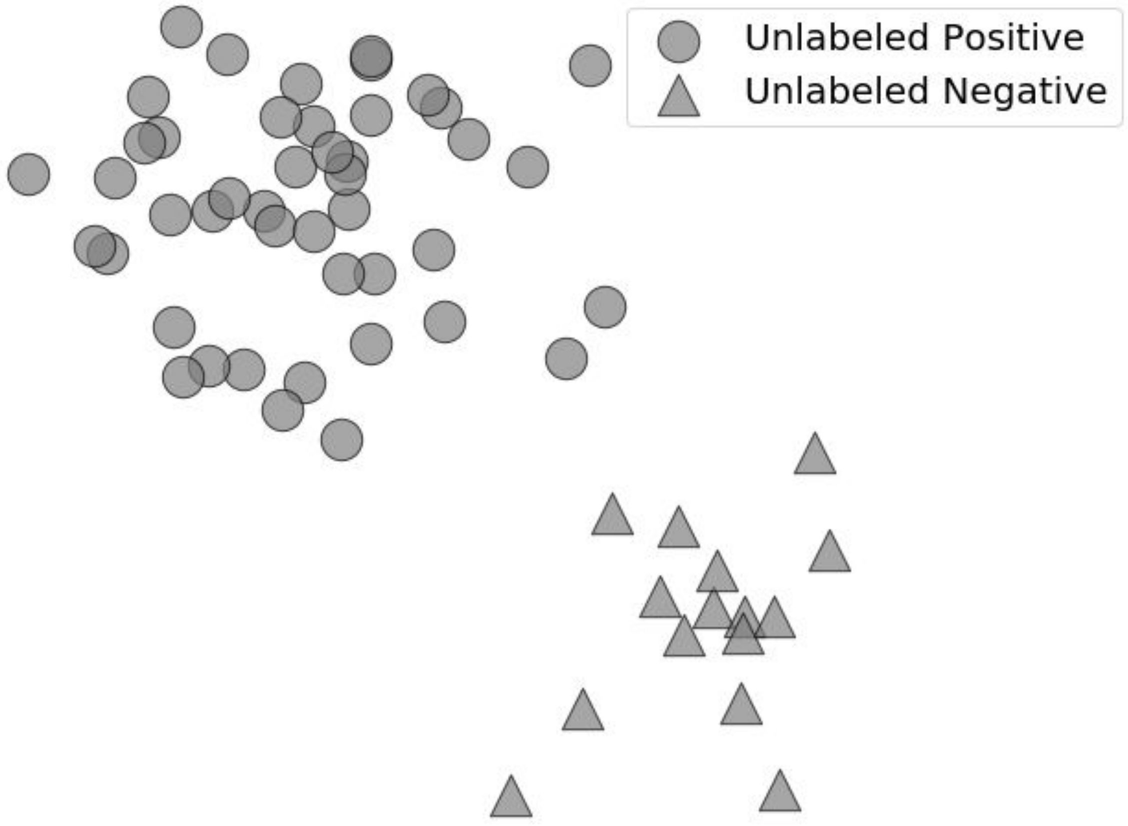
$$\mathbb{P}(labeled) = c \cdot \mathbb{P}(positive)$$

c: label frequency

Problem Setting: How to Learn from PU Data when Labels Are Acquired via Active Learning?

Active Learning Loop:

1. Train a model;
2. Select the most informative example;
3. Query the example to the user;
4. Collect the label, if positive.



Problem: *Active learning violates* ***SCAR*** *assumption*

GIF

We Make 4 Contributions

1. Formalize the problem of estimating the class prior in a *PU setting* when labels are acquired via *active learning*;

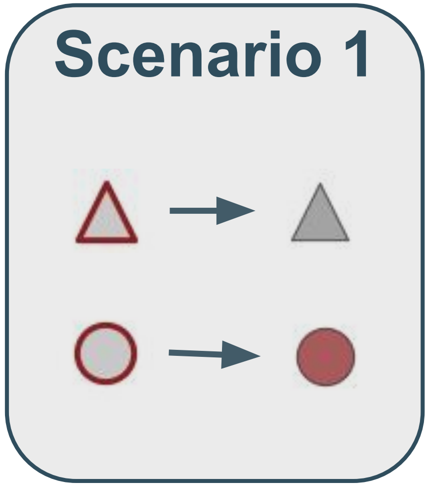
2. Propose *CAPe*, a model for estimating the *class prior* in such a scenario;

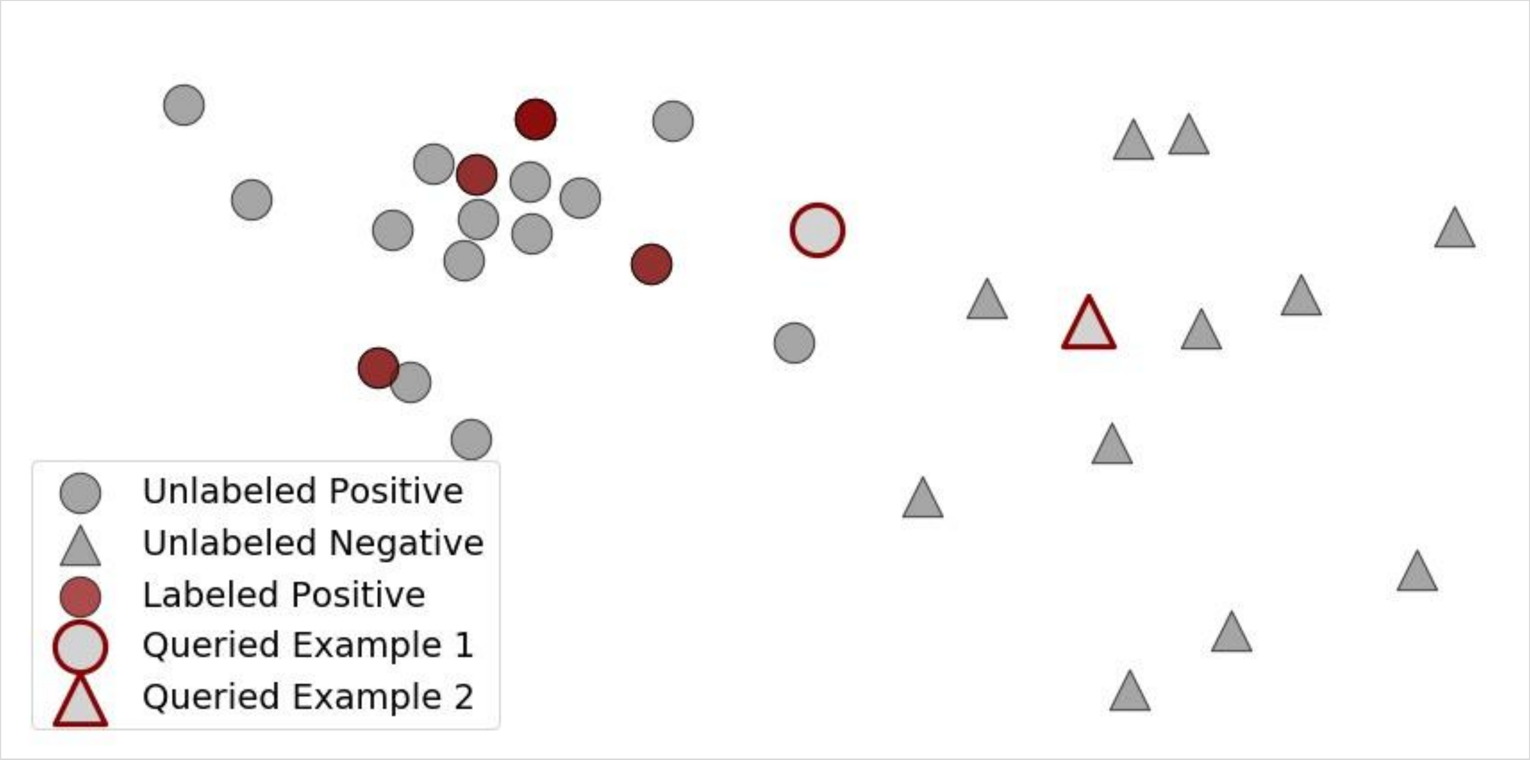
3. Prove that the estimates of the *class prior* converges to the real value;

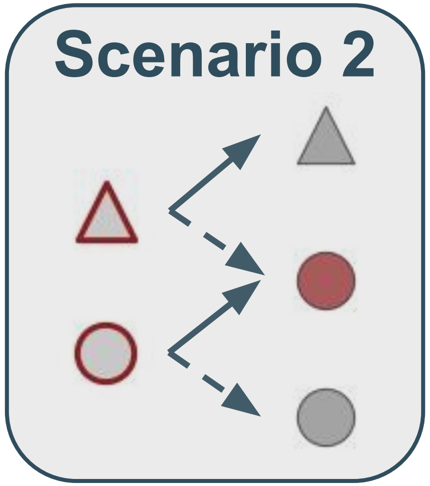
4. Evaluate empirically *CAPe* in the context of anomaly detection.

Formalizing Active Learning in a PU Setting: the User Can Only Provide Positive Labels

S1: *user only labels positive examples.*
S2: *user might label negative examples as positive or not label positive examples.*





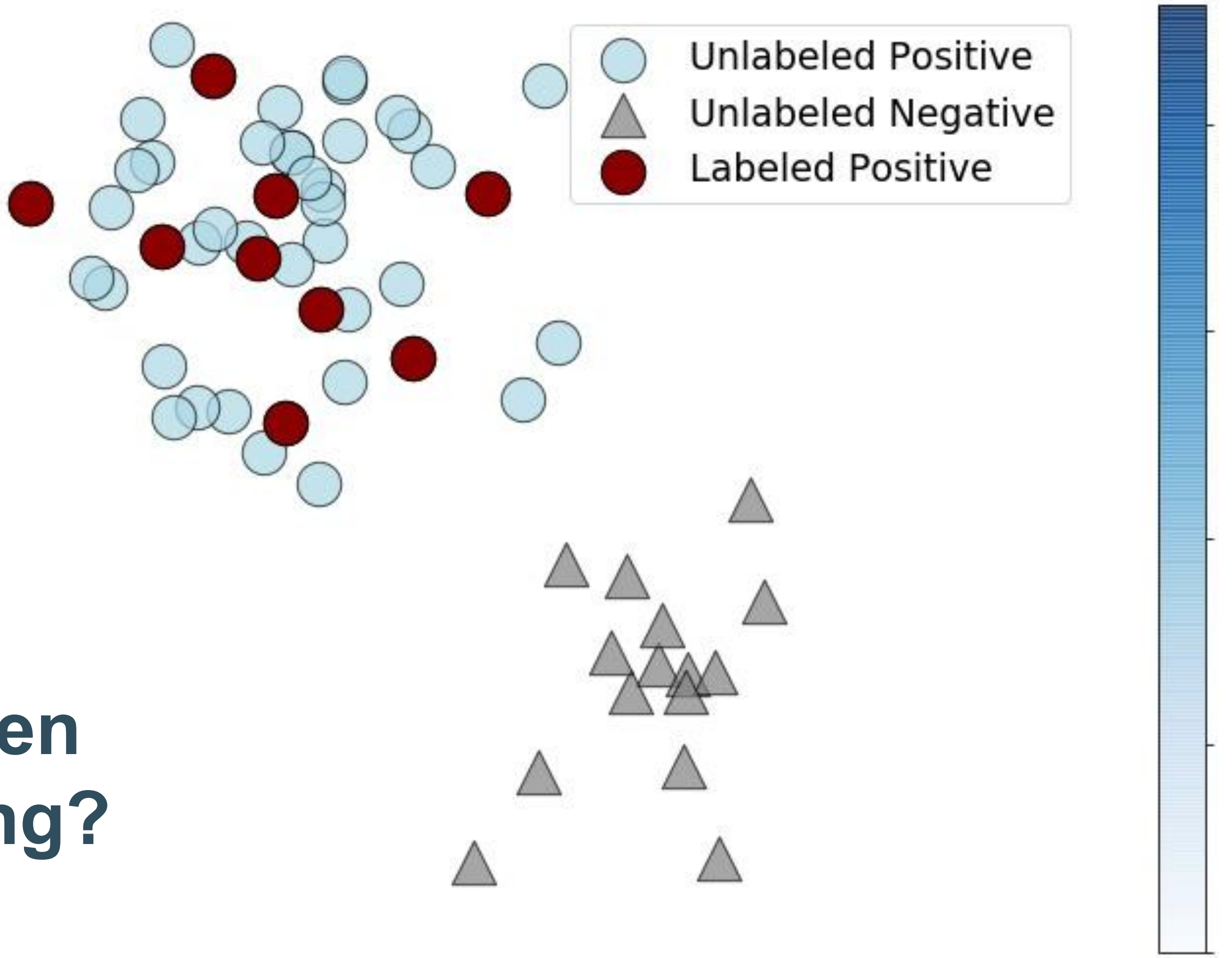


Under SCAR Assumption, Each Example Has the Same Contribution for Computing the Class Prior

Selected Completely At Random

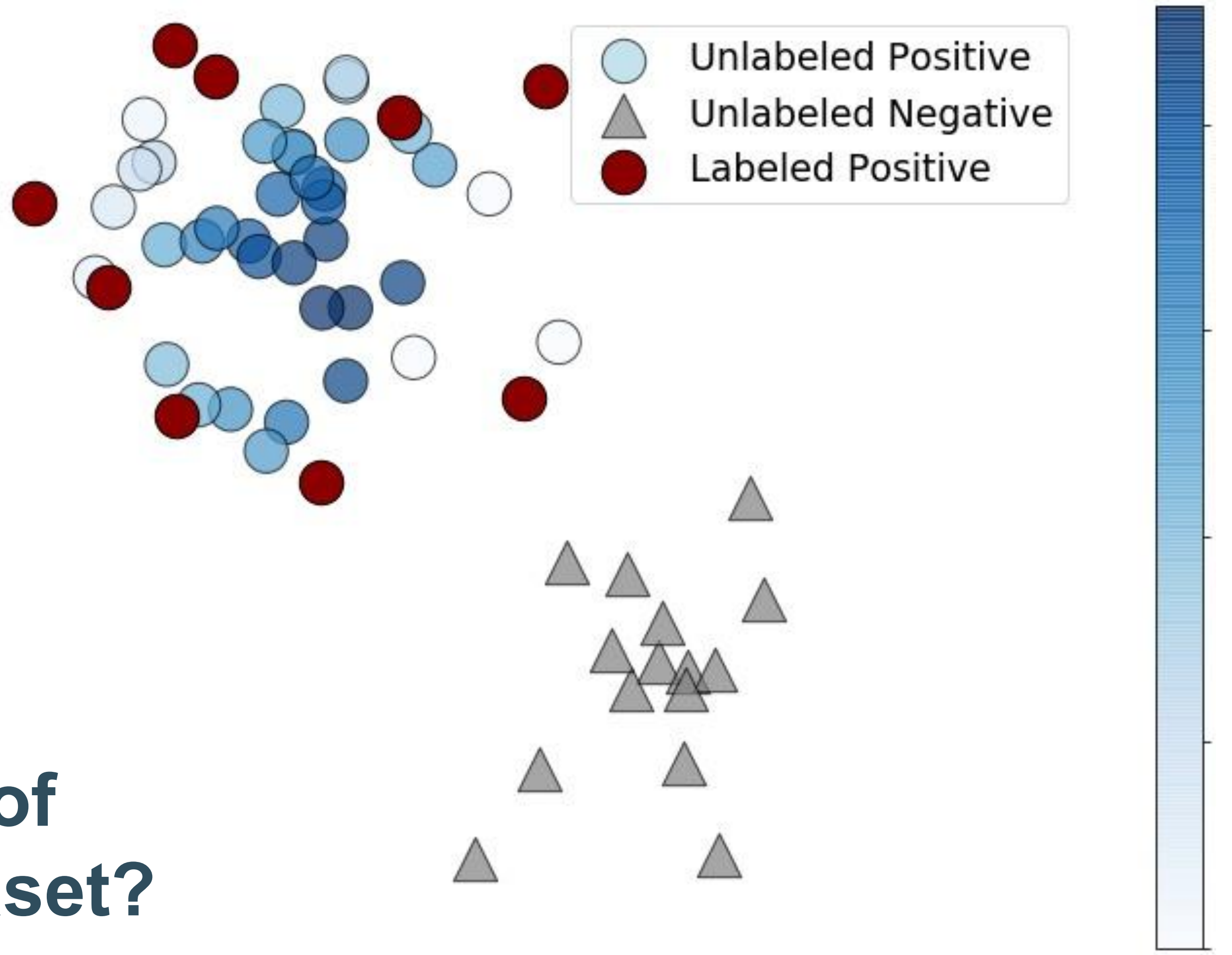
$$\mathbb{P}(labeled) = c \cdot \mathbb{P}(positive)$$

c: label frequency



Does the contribution change when collecting labels via active learning?

When Collecting Labels Via Active Learning, Examples Contribute Differently According to their Propensity Score



Propensity Score: Instance specific probability to be labeled, if positive.

High propensity score **does not** mean high contribution!

How to quantify the contribution of each positive example in the dataset?

KU LEUVEN

If the Propensity Score Was Known, We Could Compute the Class Prior as a Weighted Average

Labeled examples contributes fully

Unlabeled examples contribute their probability to be positive weighted by their propensity score

$$\mathbb{P}(positive) = \mathbb{E}_x\left[labeled\right] + \mathbb{E}_x\left[unlabeled \times \frac{\hat{y}(1-e(x))}{1-\hat{y}\,e(x)}\right]$$

- e(x) is the propensity score;
- $\hat{y} = \mathbb{P}\left(positive \,|\, x,\, e(x),\, h(x)\right) \in (0,1)$;
- h is a learning model.

How Can the Propensity Score Definition Be Adapted When Using Active Learning?

Propensity Score: Instance specific probability to be labeled, if positive.

Selected Completely At Random

- ➢ Labels collected randomly;
- ➢ The propensity score is constant;
- ➢ It equals the label frequency.

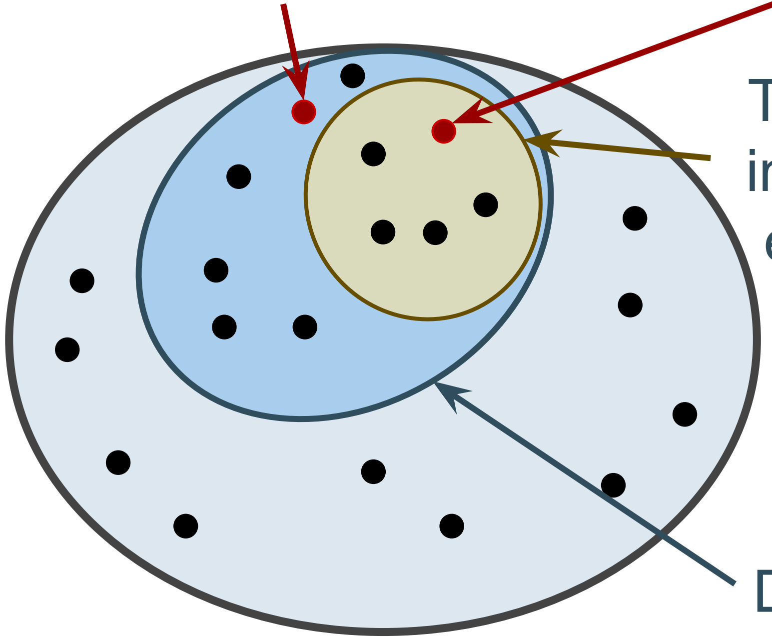
Selected by Active Learning

- ➢ Labels queried based on a strategy;
- ➢ The propensity score depends on how informative the example is;
- ➢ *It equals the probability of being queried*.

How to estimate the propensity score in practice?

Propensity Score as Proportion of Times Examples Get Queried When Drawing Different Datasets

- **Given a dataset**,
  propensity score $e(x) \in \{0, 1\}$;
- We simulate different datasets by subsampling the training set;
- We apply rules from combinatorics to avoid expensive loops and use the average rules.



Theoretical Results: Convergence Guarantees Show that CAPe Is Unbiased to the Limit

We proved that:

I. We can draw countable many samples to recover the propensity score;

II. The propensity score estimate converges to the real function when increasing the number of samples drawn from the population;

$$e_m(x) = \sum_{j=1}^{m} \sum_{\{X \subset \mathbb{R}^d,\, |X| = j,\, x \in X\}} e(x|X) \cdot d\,\mathbb{P}(X \,|\, x,\, y = 1) \overset{m \to +\infty}{\longrightarrow} e(x)$$

III. The class prior estimate converges to the real value when the propensity score converges the the real function.

$$\mathbb{P}_m(Y = 1) = \mathbb{E}_x \left[s + (1 - s) \frac{\hat{y}\,(1 - e_m(x))}{1 - \hat{y}\, e_m(x)} \right] \overset{m \to +\infty}{\longrightarrow} \mathbb{P}(Y = 1)$$

KU LEUVEN

Experimentally, We Address the Empirical Questions

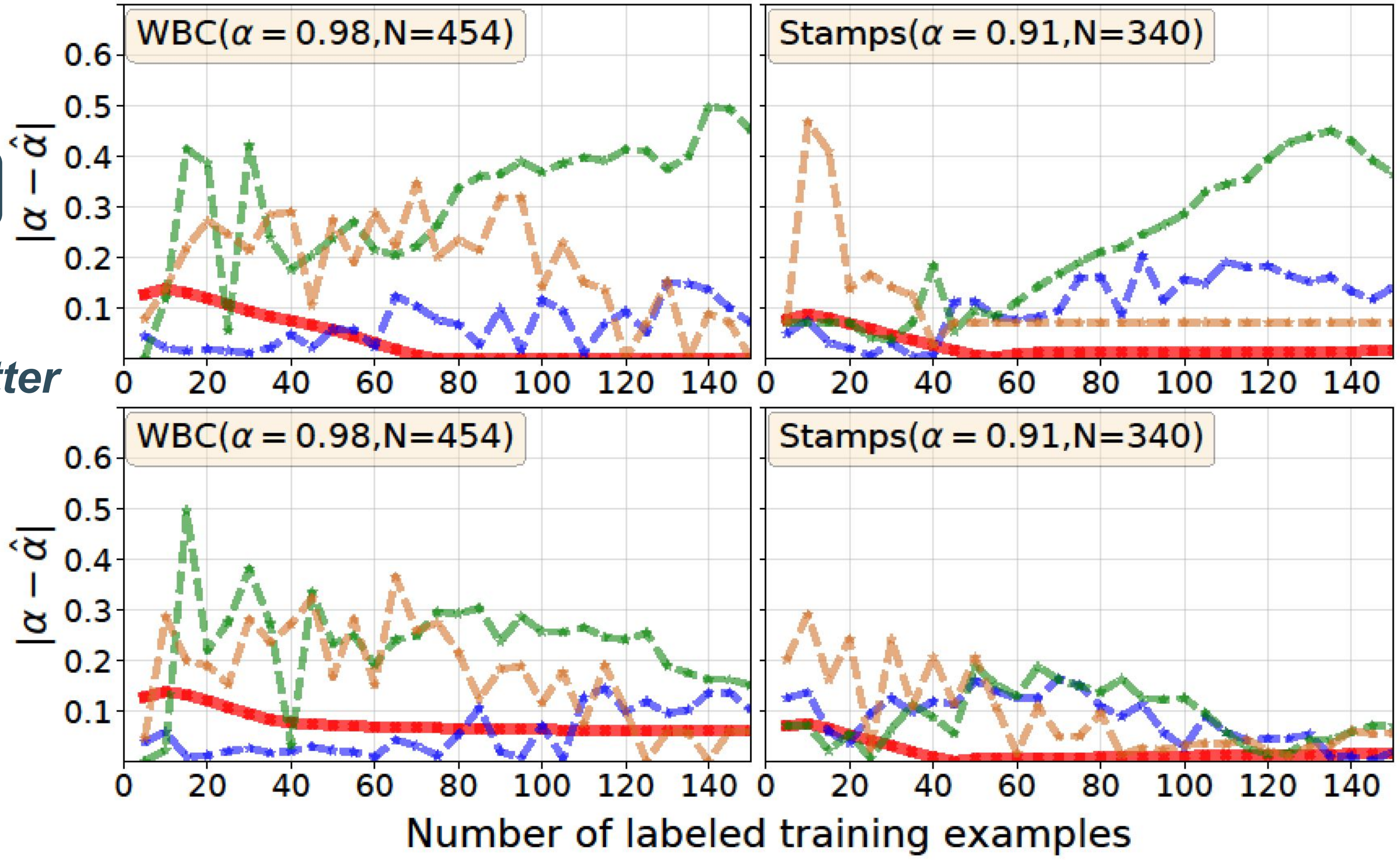
- Can *CAPe* accurately estimate the true class prior?
- How does user’s uncertainty affect *CAPe*'s ability to estimate the class prior?
- Does a more accurate estimate of the class prior improve the performance of an anomaly detector?

CAPe Recovers the Class Prior Accurately When Collecting Labels Via Active Learning

Scenario 1

Lower = Better

Scenario 2



CAPe

TIcE

Km1

Km2

In Conclusion, CAPe Accurately Recovers the Class Prior in a PU Active Setting

- We introduced a new scenario where labels are collected via active learning;
- We proposed *CAPe*, a method for estimating the class prior in anomaly detection;
- We proved that our estimates converge to the true values;
- Empirically, *CAPe* recovers *accurately* the class prior performing better than the state of the art.

All code and experiments are available online:

https://github.com/Lorenzo-Perini/Active_PU_Learning

DTAI

KU LEUVEN

Contact Us

Our Contribution:

1. Formalize the problem of estimating the class prior in a *PU setting* when labels are acquired via *active learning*;
2. Propose *CAPe*, a model for estimating the *class prior* in such a scenario;
3. Prove that the estimate of the *class prior* converges to the real value.
4. Evaluate empirically *CAPe* in the context of anomaly detection.

Lorenzo Perini, *Vincent Vercruyssen, Jesse Davis*

name.surname@kuleuven.be

IJCAI-PRICAI 2020

https://people.cs.kuleuven.be/~lorenzo.perini/

@LorenzoPerini95