Advances in Positive Unlabeled data modeling

Adam Wawrzeńczyk

PhD thesis

Advisor: prof. dr hab. Jan Mielniczuk

Institute of Computer Science,
Polish Academy of Sciences
Doctoral School of Information and Biomedical Technologies,
Polish Academy of Sciences

Warsaw, 2025

Contents

Abstract

This thesis advances the field of Positive-Unlabeled (PU) learning by addressing fundamental challenges in scenarios where only some positive examples are labeled while all negative examples remain unlabeled. The research focuses on the complex no-SCAR setting, where labeling depends on features, making classification significantly more challenging than under traditional Selected Completely At Random (SCAR) assumption.

The thesis makes several key contributions. First, it establishes fundamental incompatibilities between single sample and case control scenarios in PU learning, proving through formal analysis that cross-scenario application of methods leads to suboptimal performance. This work introduces scenario-aware variants achieving substantial improvements up to 20 percentage points while reducing overfitting.

Moreover, the thesis develops VAE-PU+OCC, a novel generative approach that enhances variational autoencoder-based PU learning through one-class classification. This method replaces problematic direct use of generated examples with identification of true positive-unlabeled examples within the original dataset, eliminating the need for risk truncation while ensuring theoretical guarantees.

Furthermore, the research introduces the first empirical procedure for False Omission Rate control in outlier detection, providing mathematical foundations and practical algorithms for controlling the proportion of undetected outliers among observations classified as inliers – critical for applications requiring high sensitivity and controlled false negative rates.

In the next step, the thesis identifies and formalizes augmented PU learning, where labeling status is available at prediction time. The optimal Bayesian decision rule d_B^{PU} is derived and integrated into VAE-PU-Bayes, achieving consistent performance improvements in classic PU task.

Finally, the work addresses label shift in *augmented* PU data, where class prior probabilities differ between training and test datasets. Theoretical analysis establishes that optimal classification requires only threshold modification, leading to practical algorithms with proven effectiveness.

The proposed methods demonstrate substantial empirical improvements across synthetic and real-world datasets, with VAE-PU-Bayes emerging as the strongest performer. This research establishes new theoretical foundations while providing practical tools that significantly advance the state-of-the-art in learning from partially labeled data under realistic, challenging conditions.

Streszczenie

Niniejsza rozprawa rozwija dziedzinę uczenia dla danych Positive-Unlabeled (PU), rozwiązując fundamentalne wyzwania w scenariuszach, gdzie tylko niektóre pozytywne przykłady są oetykietowane, podczas gdy wszystkie negatywne przykłady pozostają nieoetykietowane. Badania koncentrują się na scenariuszu no-SCAR, gdzie etykietowanie zależy od cech próbki, czyniąc klasyfikację znacznie trudniejszą niż w przypadku tradycyjnego założenia Selected Completely At Random (SCAR).

Praca przedstawia szereg osiągnięć naukowych. W pierwszej kolejności wykazano fundamentalne niekompatybilności między scenariuszami single sample i case control w uczeniu PU. Poprzez formalną analizę udowodniono, że stosowanie metod dla niepoprawnego scenariusza prowadzi do nieoptymalnej wydajności. Opracowane warianty uwzględniające scenariusz osiągają znaczące poprawy metryk (do 20 punktów procentowych) przy jednoczesnym ograniczeniu overfittingu.

Drugą kluczową innowacją jest wprowadzenie VAE-PU+OCC – podejścia generatywnego, które wzbogaca uczenie PU oparte na wariacyjnych autoenkoderach o klasyfikację jednoklasową. Metoda eliminuje problematyczne bezpośrednie wykorzystanie sztucznie wygenerowanych przykładów, zastępując je identyfikacją rzeczywistych obserwacjipositive-unlabeled w oryginalnym zbiorze danych, i usuwa konieczność obcinania funkcji ryzyka, zachowując jednocześnie gwarancje teoretyczne.

Kolejnym wkładem jest opracowanie pierwszej empirycznej procedury kontroli False Omission Rate w detekcji obserwacji odstających, dostarczając matematycznych podstaw oraz praktycznych algorytmów umożliwiających kontrolowanie proporcji niewykrytych wartości odstających wśród obserwacji zaklasyfikowanych jako normalne.

W dalszej części zidentyfikowano i sformalizowano *rozszerzone* uczenie PU, gdzie informacja o statusie etykietowania jest dostępna podczas predykcji. Wyprowadzona została optymalna reguła decyzyjna Bayesa d_B^{PU}, która została zintegrowana z metodą VAE-PU-Bayes, skutkując konsekwentnymi poprawami wydajności również w klasycznych zadaniach PU.

Ostatnim elementem badań jest analiza label shift dla rozszerzonych danych PU, gdzie prawdopodobieństwa a priori klas różnią się między zbiorem treningowym i testowym. Przeprowadzona analiza teoretyczna wykazuje, że optymalna klasyfikacja wymaga jedynie modyfikacji progu decyzyjnego, prowadząc do opracowania praktycznych algorytmów wraz z udowodnieniem ich skuteczności.

Zaproponowane metody demonstrują istotne empiryczne usprawnienia na zbiorach danych zarówno syntetycznych, jak i rzeczywistych, przy czym VAE-PU-Bayes wyłania się jako najbardziej efektywna spośród opracowanych metod. Przedstawione badania ustanawiają nowe fundamenty teoretyczne, dostarczając jednocześnie praktycznych narzędzi, które znacząco poszerzają możliwości uczenia się z danych PU w wymagających, realistycznych warunkach.

Chapter 1

Introduction

1.1 Positive-Unlabeled learning

With each passing year, innovative machine learning methods are becoming an increasingly important part of modern life. Based on the foundation of classical classification and regression problems, neural networks dominate the landscape of current technological progress. New architectures enable solving tasks previously deemed impossible in the matter of milliseconds, and dynamic research in the field introduces both new approaches as well improvements to existing methods as often as every few minutes (Maslej et al. 2024).

In any rapidly growing field, it is only natural that multiple diverse tasks arise motivated by real-world needs, often requiring special, innovative approaches. **Positive-Unlabeled learning**, for which the "**PU learning**" shorthand will be used throughout this thesis, is one of such tasks. PU learning is deeply rooted in the basic classification problems explored for decades. The key challenge differentiating it from the standard classification task is the **limited observability** of the training labels. Instead of positive and negative sets, we have access to **labeled** and **unlabeled** sets – while all of the labeled examples available during training are positive, unlabeled set is a mixture of positive and negative examples. It is also important to differentiate PU learning from some similar tasks present in the literature. **Semi-supervised learning** (Yang et al. 2023) shares a similar unlabeled set availability, but bases the learning on a subset of both positive and negative examples, which simplifies definition of the class boundary. On the other hand, **one-class classification** (Hayashi et al. 2024) aims to train a classifier based on a set of positive examples, but the additional unlabeled dataset is not available for training. Properly utilizing knowledge encoded in the unlabeled data is a key problem in PU learning, and the main reason why it can bring considerable advantages. **Learning with noisy labels** (Berthon et al. 2021) is another task which is often confused with PU learning. In this case, the training set contains both positive and negative examples, but the labels are not reliable. The difference is that in PU learning all labeled examples are positive, while in the noisy labels case, the labeled examples are positive with probability $\rho \in [0,1]$, and negative with probability $1-\rho$. Noisy label learning can be thought of as a generalization of SCAR scenario described later, as a noisy label task with $\rho = 1$ is equivalent to the SCAR PU learning. Additional thing to note is that contrary to binary classification and many other tasks, PU learning cannot be directly generalized to the multi-class classification problem, as its definition is strictly based on the binary "positive" and "negative" class terms.

Discussing PU learning would be missing the point without mentioning the context of its real-world applications. Consider the example of the detection of signaling proteins, provided by Elkan and Noto (2008). The database of identified signaling proteins contains several thousand examples (labeled examples), but the SwissProt database consisting of millions of records is expected to contain many more. As we have no access to reliable negative examples, and obtaining them is costly, using SwissProt database as a negative set is tempting, but disregards all the positive examples hidden therein. This scenario is a perfect fit for PU learning, as we can model the situation accurately by assuming that the SwissProt records are unlabeled, and enabling the detection of new signaling proteins in that set.

Another example commonly provided in PU literature is an issue of survey underreporting (Furmańczyk et al. 2022; Bekker and Davis 2020). Some surveying questions, most notably the ones which are particularly sensitive (e.g. "Have ever been abused as a child?") or negatively connoted (e.g. "Do you use drugs?"), might influence the trustworthiness of the respondents' answers. Note that also in this case, the answers often fit into the PU framework. Positive ("yes") answers to these kinds of questions do not leave much to doubt, as there is no benefit to lying in that case. However, when we inspect the negative ("no") answers, they might correspond both to people who answer truthfully, and people trying to hide the truth for various reasons, often to avoid painting themselves in the negative light. An example of such dataset is shown in Table 1.1. Traditional classifier trained on such data will fail to detect many positive cases – instead, negative answers should be recognized as unlabeled, which makes it feasible to use the PU classifier to solve such tasks.

Yet another example of the PU learning application is ecology (Ward et al. 2009). Consider occurrence of rare animal species in various ecosystems. Detected animal sighting is enough to find the positive examples; note, however, that absence of a sighting in a particular region is not as reliable – the animal, due to its potential rarity, can still live in such area even if it was not sighted yet. This is another case of unreliable example labels leading to natural PU dataset occurrence. There is no lack of PU learning applications in other domains, such as genetics, recommendation systems and many more (Bekker and Davis 2020; Chen et al. 2021; Zhou, Xu, et al. 2021; Zhao, Pang, et al. 2021; Zhou, Lu, et al. 2022). They share common characteristics – obtaining negative labels is either very hard or the negatives are not reliable;

Table 1.1: Example of a PU dataset. The feature vector constitutes of four variables: "Age", "Weight", "Gender" and "Education". The classification goal is to predict the value of the variable "Drug use". However, this variable is not directly observable – the data is collected as part of a nationwide survey. Instead of the class variable, the training dataset will contain the "Survey result" column. Classifier based on it directly will fail to detect many drug use cases.

Age	Gender	Education	Drug use	Survey result	Reason to lie
20	female	higher	No	No	-
50	male	primary	Yes	Yes	-
18	female	secondary	Yes	No	Young woman ashamed of drug use
25	male	higher	No	No	-
70	female	secondary	No	No	-
45	female	primary	Yes	Yes	-
40	male	higher	Yes	No	Well off businessman worried about social stigma of drug use

and when presented with such dataset, it is easy to fall into a trap of unknowingly training a traditional classifier, underestimating the occurrence of true positive examples in the real population and leading to wrong predictions and conclusions.

1.2 Research aim and thesis structure

The research presented in this thesis is motivated by two fundamental challenges in the field of Positive-Unlabeled learning: the prevalent mistakes commonly made when handling PU data across different scenarios, and the lack of powerful methods available for the challenging no-SCAR PU learning setting. The primary research goal is twofold: first, to conduct a comprehensive exploration of PU learning scenarios, identifying their unique characteristics and developing scenario-aware methodologies; and second, to advance the state-of-the-art in no-SCAR PU learning through the development of novel deep learning approaches that can effectively handle the complex dependencies between labeling mechanisms and feature representations.

The thesis systematically addresses these challenges by developing a series of interconnected methodological contributions. The exploration begins with a rigorous analysis of scenario-specific behaviors in PU learning, revealing fundamental incompatibilities that have been largely overlooked in existing literature. Building upon these insights, the research transitions to the development of sophisticated deep learning architectures specifically designed for no-SCAR settings, where traditional assumptions about random labeling mechanisms fail. The proposed methods leverage advanced generative modeling techniques, one-class classification principles, and novel risk control mechanisms to create robust solutions for real-world PU learning scenarios.

The thesis will sequentially cover various approaches developed during exploration of those topics. Each chapter constitutes a self-contained unit building upon the foundations laid in the preceding parts of the thesis, each introducing a new problem, discussing the proposed solution and verifying it via practical experiments.

First part of the thesis will explore the twin single sample and case control scenarios characterizing PU datasets. Motivated mostly by confusion in that regard shown even by highly skilled researchers, in Chapter 3 both scenarios will be explored in depth, resulting in unveiling the scenario-aware version of widely used uPU and nnPU methods introduced by Kiryo et al. (2017), and discussing the importance of the correct scenario identification in PU learning tasks.

Starting from Chapter 4 the focus shifts to the no-SCAR PU learning. There, we will introduce the autoencoder architecture and the no-SCAR VAE-PU model, used as a base for the further research presented in the thesis. A series of improvements and modifications results in the VAE-PU+OCC model based on the one-class classification approach, which addresses the deficiencies identified in the VAE-PU baseline by reducing the impact of the imperfectly generated examples in the learning process.

Further chapters explore various improvements centered around the VAE-PU+OCC. In Chapter 5, methods other than one-class classification in this context are examined. Inspired by the classic Benjamini-Hochberg method for False Discovery Rate control, a novel False Omission Rate control method, focused on the opposite non-discovered set purity, will be introduced. Its effectiveness will be assessed not only in the context of pure outlier detection tasks, but also as a component of the VAE-PU+FOR variant optimized for selected set purity.

Chapter 6 starts off by identifying a previously unexplored augmented PU learning task, utilizing the knowledge of the example label at prediction time, and which is motivated by its real-world application analysis. Following is the exploration of the adaptations necessary for the PU learning methods to enable training under augmented PU scenario. An important milestone is identification of augmented PU subtask in a standard PU learning scenario, which culminates in proposing a refinement of the previous proposed methods – VAE-PU-Bayes model.

The last significant part of the thesis focuses on the unique challenges presented by the label shift occurring in the PU datasets. In Chapter 7, the thesis focuses on augmented PU classification threshold modification, which is proven to suffice in order to handle the change of class prior in the test dataset. The modification results in yet another VAE-PU variant, which is then evaluated in the final array of experiments.

The thesis closes by summarizing the research in Chapter 8. Summary of the thesis results and bibliography is followed by appendices outlining theoretical justifications of the proposed methods.

The thesis content reflects a series of publications in scientific journals and conference appearances. Discussion of scenario differences and their consequences in Chapter 3 is based on the paper published in the Fundamenta Informaticae (Mielniczuk and Wawrzeńczyk 2025b). Research topics presented in Chapters 4 and 6 regarding solving no-SCAR and augmented no-SCAR PU problems were presented as a part of ECAI 2023 (Mielniczuk and Wawrzeńczyk 2023) and ECAI 2024 (Mielniczuk and Wawrzeńczyk 2024) programme, respectively, while the FOR control method discussed in Chapter 5 was first introduced in the proceedings of the ICCS 2023 conference (Wawrzeńczyk and Mielniczuk 2023b). The content of Chapter 7,exploring the augmented PU learning under the lens of label shift, are are the topic of the forthcoming paper in AMCS journal (Mielniczuk and Wawrzeńczyk 2025a). It's also worth mentioning an earlier paper related to the subject of this thesis based on the earlier M.Sc. thesis. Focusing on optimization challenges of the joint model used in the SCAR PU problems, the paper was also published (Wawrzeńczyk and Mielniczuk 2022).

Chapter 2

Preliminaries

2.1 Basic PU learning terminology

Let $X = (x_1, ..., x_p)^T \in \mathbb{R}^p$ be a random variable corresponding to a **feature vector**, and $Y \in \{-1, 1\}$ be a random variable denoting a true class indicator; $Y = 1$ corresponds to a **positive** example from class **P**, and $Y = -1$ – **negative** example from class **N**. In the following, $\boldsymbol{P_{XY}}$ will denote **joint probability distribution** of (X, Y), and $\boldsymbol{P_{X|Y=1}}$ ($\boldsymbol{P_{X|Y=-1}}$) – the **probability distribution** of **positive** (**negative**) examples. We assume that positive examples occur in the general population with a **class prior probability** $\boldsymbol{\pi} = P(Y = 1)$. Variable $S \in \{0, 1\}$ is an indicator of an example being **labeled** ($S = 1$, labeled set $\boldsymbol{L}$) or **unlabeled** ($S = 0$, unlabeled set $\boldsymbol{U}$). The core PU learning assumption is that only positive examples can be labeled, i.e. $P(Y = 1|S = 1, X) = 1$. There are no such restrictions on the unlabeled dataset – it is a mixture of both positive and negative examples in any proportion. The main goal of the PU learning task is unchanged from the traditional classification problem – obtaining binary posterior distribution of Y given $X = x$, i.e. $y(x) = P(Y = 1|X = x)$; however, contrary to standard classification where we observe examples drawn from the distribution (X, Y), observed PU data comes from (X, S) distribution.

To describe the distribution of positive examples between classes, it is useful to define the notion of propensity score. **Propensity score** $\boldsymbol{e(x)}$ describes the probability of a randomly chosen positive example being labeled:

$$e(x) := P(S = 1|Y = 1, X = x). \tag{2.1}$$

In the PU learning literature it is common to adopt the Selected Completely At Random (**SCAR**) assumption (Bekker and Davis 2020). It decouples the propensity score from the example properties, making it constant:

$$e(x) = P(S = 1|Y = 1, X = x) = P(S = 1|Y = 1) =: c. \tag{2.2}$$

The constant $\boldsymbol{c}$ appearing on the right side of the above equation is another of the key characteristics of the PU data; it is called the **label frequency** and describes the prevalence of labeled examples in the positive population.

While SCAR assumption greatly simplifies the problem, it is easy to notice its shortcomings. The main issue is that SCAR data is extremely rare in practice; many problems exhibit strong preferentiality in their labeling. To give a few examples, older and more educated people are more likely to report their illnesses, some properties of genes might make their connection to certain diseases easier to detect, and sightings of certain animals might be more common than others. The other problem is that SCAR-based methods show subpar performance when applied to non-SCAR data, as shown in several papers, e.g. Bekker, Robberechts, and Davis (2019) and Na et al. (2020).

The SCAR assumption issues coupled with the emergence of new neural network architectures led to the recent dynamic development of the **no-SCAR** PU learning methods. Alternatively, they operate under the Selected At Random (**SAR**) assumption, which, in its essence, preserves the dependence of propensity score on the example features, i.e. $e(x) = P(S = 1|Y = 1, X = x)$. In this case, the assumption corresponds to the premise that X contains all predictors on which labeling process may possibly depend. Such task is naturally much more difficult than the SCAR equivalent.

When discussing PU learning, it quickly becomes apparent that obtaining PU datasets can be a result of two distinct processes. In a **single sample** scenario (abbreviated to **SS**; also called a single training set or censoring scenario) we assume that there is some unknown distribution $P_{X,Y,S}$ such that $(X_i, Y_i, S_i), i = 1, \ldots, n$ are independent and identically distributed (iid) random variables drawn from it. In contrast, in **case control** (**CC**) scenario we observe two example sets, the first $X_1, \ldots, X_{n_1}$ pertaining to the positive class $\boldsymbol{P}$ (distributed according to $P_{X|Y=1}$) and the second $X_{n_1+1}, \ldots, X_{n_1+n_2}$ drawn from a general population being the mixture of distributions $P_X = \pi P_{X|Y=1} + (1-\pi)P_{X|Y=-1}$. Note that while the strictly probabilistic meanings of propensity score the derived SCAR and SAR concepts are specific to single sample scenario, we can define their case control equivalents using the same equations if we treat the two example sets as a single set, and using $S = 1$ for labeled set and $S = 0$ for the unlabeled one rather than treating it as a random variable. For such correspondence distribution of labeled elements in CC scenario equals distribution of labeled elements in SS scenario. The dependencies between classes in such a combined set are different than the dynamics of the single sample dynamics, which requires extra caution; discussion of those differences will be expanded in Chapter 3. We stress that no-SCAR scenario well be meant here in SS setting.

Another important aspect of working with PU data is the estimation of class prior $\boldsymbol{\pi}$. Knowledge of π is not available based on the learning set, and many algorithms require its estimation in order to function properly. Exploration of this area is quite extensive, with many diverse approaches such as general solution using decision tree induction (Bekker and Davis 2018a), or, with additional assumptions, non-standard classifiers, partial matching, ROC or kernel embeddings (Bekker and Davis 2020). Moreover, there are also many methods which estimate π during the training, or where it can be extracted as the solution byproduct (Furmańczyk et al. 2022; Gong et al. 2021), however, they require additional assumptions which ensure identifiability of π. The other common approach is to assume knowledge of π in the algorithm. This assumption is realistic in many cases where we can estimate the prevalence of a particular target (e.g. the disease) in the general population. This is also the approach which will be used throughout this thesis, assuming that class prior is either known or accurately estimated using one of the existing algorithms mentioned above.

2.2 Single sample scenario – basic properties

As the majority of this thesis focuses on the single sample scenario, we now take a look at some simple properties of this framework defined in the following Lemmas. First, Lemma 2.1 focuses on the interplay of the posteriors $s(x)$ and $y(x)$.

Lemma 2.1. *In single sample scenario, we have*

$$s(x) = e(x)y(x) \tag{2.3}$$

Proof. By using the definitions and the PU assumption $P(S = 1, Y = 1|X) = P(S = 1|X)$, we have

$$\begin{aligned} s(x) &= P(S = 1|X = x) \\ &= P(S = 1, Y = 1|X = x) \\ &= P(S = 1|Y = 1, X = x)P(Y = 1|X = x) \\ &= e(x)y(x) \end{aligned}$$

□

Lemma 2.2 shows that the labeled examples in SCAR scenario are generated from the same distribution as the observations from the positive class.

Lemma 2.2. *In single sample scenario, under SCAR we have* $P_{X|S=1} = P_{X|Y=1}$.

Proof. This easily follows after noting that by switching the conditioning we have

$$P(X=x|S=1)=\frac{P(S=1|X=x)P_X(X=x)}{P(S=1)}$$

and that in view of the fact $S=1$ implies $Y=1$ and (2.2) we have

$$\begin{aligned}P(S=1|X=x)=&P(S=1|Y=1,X=x)P(Y=1|X=x)\\=&P(S=1|Y=1)P(Y=1|X=x)=\frac{P(S=1)}{P(Y=1)}P(Y=1|X=x).\end{aligned}$$

Plugging in the formula for $P(S=1|X=x)$ into the first equation we have $P(X=x|Y=1)$ after switching back the conditioning. □

Lemma 2.3 focuses on the conditional independence of X and S for a given class, in contrast to the dependence between on S on Y.

Lemma 2.3. *In single sample scenario, under SCAR we X and S are* ***conditionally independent*** *($\perp\!\!\!\perp$) given Y:*

$$X \perp\!\!\!\perp S|Y \tag{2.4}$$

but S is not independent of Y:

$$S \not\perp\!\!\!\perp Y \tag{2.5}$$

Proof. Proving that S is not dependent of X for a given Y is straightforward. For positive examples, by definition of SCAR

$$P(S=1|Y=1,X=x)=P(S=1|Y=1),$$

and consequently

$$P(S=0|Y=1,X=x)=1-P(S=1|Y=1,X=x)=1-P(S=1|Y=1)=P(S=0|Y=1),$$

which implies $X \perp\!\!\!\perp S|Y=1$. For $Y=-1$

$$P(S=1|Y=-1,X=x)=0=P(S=1|Y=-1),$$

which concludes the proof.

The proof of the second part of the lemma is also trivial, as $P(S=0|Y=-1)=1$, but $P(S=0|Y=1)=1-c$ is not equal to 1 unless $c=0$ and thus all examples in the dataset are labeled as negative (and datasets like this make it impossible to learn any kind of binary classifier). □

2.3 Empirical risk minimization

Let $\boldsymbol{g(x)} : \mathbb{R}^p \to \mathbb{R}$ be a **classification function** with the corresponding **classifier** defined as $\boldsymbol{d(x)} = 2\mathbb{I}\{g(x) \geq 0\} - 1$ (i.e. classifying to positive class $Y = 1$ for $g(x) \geq 0$ and to the negative one $Y = -1$ in the opposite case). Moreover, $l : \mathbb{R} \times \{-1, 1\} \to \mathbb{R}^+$ is a **loss function** (usually nonincreasing and convex), with $l(g(x), y)$ standing for the loss incurred for classification function $g(x)$, when the true class indicator is y. From now on, abusing the notion slightly, we consider the losses of the form $l(g(x), y) := l(yg(x))$, where l is defined now on $\mathbb{R}$. An ideal loss function l aims to minimize the number of misclassifications, which can be also described as a 0–1 loss $l_{0-1}(t) = \mathbb{I}\{t < 0\}$. As 0–1 loss function is neither differentiable nor convex, it cannot be used directly in the process of optimization; some commonly used approximations are:

- logistic (sigmoid) loss $l_{\log}(t) = \log(1 + e^{-t})$,
- inverse sigmoid loss $l_{\text{inv}}(t) = 1/(1 + e^{-t})$,
- exponential loss $l_{\exp}(t) = e^{-t}$.

Empirical Risk Minimization (**ERM**) approach originates from statistical decision theory DeGroot (2004). The objective is to minimize empirical version of the expected loss $R_{XY}(g) = \mathbb{E}\ l(Yg(X))$ over some family of functions which is given e.g. as an output of neural network with a fixed architecture. For a more detailed overview about ERM approach, see e.g. Hastie, Tibshirani, and Wainwright (2015) for a general introduction, and Coudray et al. (2023) for a recent ERM approach to biased PU data. The obtained minimizer g^* is of the form

$$g^* = \arg\min_{g \in G} R(g), \tag{2.6}$$

where G is the considered family of classification functions – or, in terms of derivative (for the strictly convex loss functions considered above)

$$g^*(x) : \frac{\partial}{\partial g(x)} \mathbb{E}_{Y|X=x}(g^*(x)) = 0.$$

In the case of logistic loss, zero of the derivative of conditional risk above for an arbitrary value of x is attained at (see Appendix B.1 for the full derivation):

$$g^*_{\log}(x) = \log \frac{P(Y = 1|x)}{P(Y = -1|x)},$$

thus giving solution to 2.6. Note that $g^*_{\log}(x)$ is a nondecreasing function of odds, and the $d(x)$ classification rule based on follows the **Bayes rule**, classifying the example x to positive class if it's more likely to belong to it over negative class ($P(Y=1|x) \geq P(Y=-1|x)$), as:

$$\begin{gathered}
g^*_{\log}(x) \geq 0 \\
\log \frac{P(Y=1|x)}{P(Y=-1|x)} \geq 0 \\
\frac{P(Y=1|x)}{P(Y=-1|x)} \geq 1 \\
P(Y=1|x) \geq P(Y=-1|x)
\end{gathered}$$

As a side note, the Bayes rule in the last line above is equivalent to the prevalent classification rule based on posterior $y(x)$, classifying to positive class if $P(Y=1|x) = y(x) \geq 0.5$.

Similarly, for the exponential loss, zero of the derivative of conditional risk for an arbitrary value of x is attained at (see Appendix B.2 for the full derivation):

$$g^*_{\exp}(x) = \frac{1}{2} \log \frac{P(Y=1|x)}{P(Y=-1|x)}$$

which, differing only by the constant factor of $\frac{1}{2}$ in front of the logarithm, shares the properties of logistic loss. In contrast, it is proven in the Appendix B.3 that risk for the inverse sigmoid loss does not have a stationary point.

In the derivations until now, our assumed risk function directly referenced Y, which is not available under the PU framework. Theorem 2.4 proposes an alternative formulation of the risk function, which is more suitable for the PU learning tasks. It holds for the no-SCAR scenario, and thus SS setting is assumed here.

Theorem 2.4. *The general no-SCAR PU risk equals:*

$$\begin{aligned}
R(g) =& P(S=1)\, \mathbb{E}_{X|S=1}\, l(g(x)) \\
&+ P(Y=1, S=0)(\mathbb{E}_{X|Y=1,S=0}\, (l(g(x)) - l(-g(x))) \\
&+ P(S=0)\, \mathbb{E}_{X|S=0}\, l(-g(x))
\end{aligned} \tag{2.7}$$

Proof. By definition of $R(g)$ we have

$$\begin{aligned}
R(g) =& \mathbb{E}_{X,Y}\, l(Yg(x)) \\
=& \mathbb{E}_{X,Y=1}\, l(g(x)) \\
&+ \mathbb{E}_{X,Y=-1}\, l(-g(x)) \\
=& \mathbb{E}_{X,S=1}\, l(g(x)) \\
&+ \mathbb{E}_{X,S=0,Y=1}\, l(g(x)) \\
&+ \mathbb{E}_{X,S=0}\, l(-g(x)) \\
&- \mathbb{E}_{X,S=0,Y=1}\, l(-g(x)) \\
=& P(S=1)\, \mathbb{E}_{X|S=1}\, l(g(x)) \\
&+ P(Y=1,S=0)\, \mathbb{E}_{X|Y=1,S=0}\, l(g(x)) \\
&- P(Y=1,S=0)\, \mathbb{E}_{X|Y=1,S=0}\, l(-g(x)) \\
&+ P(S=0)\, \mathbb{E}_{X|S=0}\, l(-g(x)) \\
=& P(S=1)\, \mathbb{E}_{X|S=1}\, l(g(x)) \\
&+ P(Y=1,S=0)\, \mathbb{E}_{X|Y=1,S=0}\, l(g(x)) - l(-g(x)) \\
&+ P(S=0)\, \mathbb{E}_{X|S=0}\, l(-g(x)).
\end{aligned}$$

□

Note that $-g(x)$ in the above theorem corresponds to classification function which classifies to the opposite class than $g(x)$.

Under SCAR assumption, the risk function used for ERM minimization can be simplified. First, we will prove Theorem 2.5, handling both single sample and case control scenarios.

Theorem 2.5. *Under SCAR assumption, the following general formula for the risk $R(g)$ holds regardless of the scenario considered:*

$$\begin{aligned}
R(g) =& \pi\, \mathbb{E}_{X|S=1}\, l(g(x)) \\
&+ \mathbb{E}_X\, (l(-g(x))) \\
&- \pi\, \mathbb{E}_{X|S=1}\, (l(-g(x)))\,.
\end{aligned} \tag{2.8}$$

Proof. This immediately follows after noticing that

$$\mathbb{E}_X\, (l(-g(x))) = \pi\, \mathbb{E}_{X|Y=1}\, l(-g(x)) + (1-\pi)\, \mathbb{E}_{X|Y=-1}\, l(-g(x)).$$

which leads to

$$\begin{aligned} R(g) =& \pi\, \mathbb{E}_{X|Y=1}\, l(g(x)) \\ &+ \mathbb{E}_X\, (l(-g(x))) \\ &- \pi\, \mathbb{E}_{X|Y=1}\, (l(-g(x)))\,. \end{aligned} \tag{2.9}$$

We obtain the thesis applying Lemma 2.2 to the resulting expression in SS scenario, and remembering that labeled population is drawn from the positive distribution in CC scenario. □

For single sample scenario we prove a formula for $R(g)$ which relies solely on $P_{X|S=1}$ and $P_{X|S=0}$ and which will be used later in the thesis to provide additional insight how procedures designed for CC case perform in SS case. This is given by the following Theorem which is a special case of representation of $R(g)$ in (Bekker and Davis 2020), p. 23, line 3. We note that although Theorem 2.6 is proved for SS framework, it formally coincides with Theorem 2.5 when $\mathbb{E}_{X|S=0}$ is replaced by $\mathbb{E}_X$, $P(S=0)$ by 1 and $P(Y=1, S=0)$ by π.

Theorem 2.6. *Under SCAR assumption in single sample scenario, $R(g)$ equals:*

$$\begin{aligned} R(g) =& \pi\, \mathbb{E}_{X|S=1}\, l(g(x)) \\ &+ P(S=0)\, \mathbb{E}_{X|S=0}\, l(-g(x)) \\ &- P(S=0, Y=1)\, \mathbb{E}_{X|S=1}\, l(-g(x)) \end{aligned} \tag{2.10}$$

Proof. This follows directly from Theorem 2.5. Note that using Lemma 2.2 in single sample scenario we have

$$\mathbb{E}_{X|Y=1}\, l(g(X)) = \mathbb{E}_{X|S=1}\, l(g(X)).$$

and because of $P(S=1, Y=1) = P(S=1)$ we have

$$P(S=0, Y=1) = P(Y=1) - P(S=1) = \pi - P(S=1).$$

Thus by transforming Theorem 2.6, we have

$$\begin{aligned} R(g) =& \pi\, \mathbb{E}_{X|S=1}\, l(g(x)) + P(S=0)\, \mathbb{E}_{X|S=0}\, l(-g(x)) - P(S=0, Y=1)\, \mathbb{E}_{X|S=1}\, l(-g(x)) \\ =& \pi\, \mathbb{E}_{X|S=1}\, l(g(x)) + P(S=0)\, \mathbb{E}_{X|S=0}\, l(-g(x)) - (\pi - P(S=1))\, \mathbb{E}_{X|S=1}\, l(-g(x)) \\ =& \pi\, \mathbb{E}_{X|S=1}\, l(g(x)) + P(S=0)\, \mathbb{E}_{X|S=0}\, l(-g(x)) + P(S=1)\, \mathbb{E}_{X|S=1}\, l(-g(x)) - \pi\, \mathbb{E}_{X|S=1}\, l(-g(x)) \\ =& \pi\, \mathbb{E}_{X|Y=1}\, l(g(x)) + \mathbb{E}_{X,S=0}\, l(-g(x)) + \mathbb{E}_{X,S=1}\, l(-g(x)) - \pi\, \mathbb{E}_{X|Y=1}\, l(-g(x)) \\ =& \pi\, \mathbb{E}_{X|Y=1}\, l(g(x)) + \mathbb{E}_X\, l(-g(x)) - \pi\, \mathbb{E}_{X|Y=1}\, l(-g(x)) \end{aligned}$$

which is precisely the formula from more general Theorem 2.5, concluding the proof. □

Remark 2.7. *Define $\widetilde{S} = 2S - 1$ (i.e. transforming $S \in \{0,1\}$ into $\widetilde{S} \in \{-1,1\}$). As a side note we remark that in the case of logistic loss $l(s) = \log(1 + e^{-s})$ for which $l(s) - l(-s) = -s$, risk from Equation* (2.10) *for linear classification function $g(x) = \beta^T x$ simplifies to*

$$R(g) = \mathbb{E}\left(l(Y\beta^T X)\right) = \mathbb{E}\left(l(\widetilde{S}\beta^T X)\right) - P(Y = 1, \widetilde{S} = -1)\beta^T \mathbb{E}_{X|Y=1,\widetilde{S}=-1} X,$$

for which correction term $\mathbb{E}\left(l(Y g(X))\right) - \mathbb{E}\left(l(\widetilde{S} g(X))\right)$ is linear function of β.

2.4 Existing PU learning methods

The first influential paper discussing PU learning problem in the context of identifying signaling proteins in SwissProt database was a paper by Elkan and Noto (2008). Under the single sample scenario, the authors proposed several theoretical results for PU data, including the simplified Lemma 2.1 formulation, and a weighted classifier method handling the PU data. For case control scenario, the research originated from Ward et al. (2009) paper, which proposed EM algorithm for logistic model in this context. Since then the PU learning problem has been extensively studied in the literature, with a wide range of algorithms proposed to tackle it. PU learning algorithms is usually broadly divided into three categories: **two-step** techniques, **biased** learning and **class-prior-based** methods (Bekker and Davis 2020). The groups are not mutually exclusive, and many methods can be classified into more than one category, e.g. there exist biased learning methods which utilize the class prior. Let us mention in passing that the term "non-traditional classifier" is used in this context, and corresponds to a method which simply treats all unlabeled data as negative examples and which is then subsequently refined.

Two-step techniques first aim to identify reliable negative examples in the unlabeled set (and, optionally, additional reliable positives). Then, as the second step, a classifier is trained using a traditional supervised method (or semi-supervised method capable of handling the remaining portion of the unlabeled set). Applying this approach is based on somewhat restrictive assumption that positive examples from the unlabeled set are similar to the labeled ones, while negative ones are significantly different. Despite that, the two-step techniques found a widespread usage especially in the natural text processing domain (Bekker and Davis 2020). An example of such method is S-EM (Liu, Lee, et al. 2002), where the authors introduce the concept of the spies – labeled examples added to the unlabeled set; using a classifier considering unlabeled examples as negative, all the examples with posterior lower than all of the spies are considered as reliable negatives. Other examples of two-step methods include Roc-SVM (Li and Liu 2003), PEBL (Yu, Han, and Chang 2003), A-EM (Li and Liu 2005), MCLS (Chaudhari and Shevade 2012) and PGPU (He et al. 2018), each proposing a different combination of approaches to both of the steps.

Biased learning methods consider the unlabeled set as a noisy negative set, with the noise being constant, i.e. $P(S = 0|Y = 1, X)$ is not dependent on X (which is equivalent to the SCAR assumption). The main variants of this approach include introducing an asymmetric penalty for the incorrectly classified examples (Ke et al. 2012; Liu, Dai, et al. 2003), as well as defining weights approximating the probability of being negative for the unlabeled examples (Liu, Shi, et al. 2005). These approaches can be combined with bagging to improve the performance for very noisy data (Claesen et al. 2015; Mordelet and Vert 2010), with a big advantage of being able to use any classifier as a base learner, including SVM or decision trees. Another popular approach are hyperplane optimization methods, including RankSVM (Sellamanickam, Garg, and Selvaraj 2011), Biased Twin SVMs (Xu, Qi, and Zhang 2014), NPSVM (Zhang, Ju, and Tian 2014) and LUHC (Shao et al. 2015).

Class-prior-based methods, in literature commonly used together with the SCAR assumption, utilize the knowledge of the class prior π to simplify the PU learning problem. A straightforward approach is to postprocess the classifier output: as a direct consequence of Lemma 2.1, in SCAR scenario we have $s(x) = cy(x)$, and c is known due to $c = P(S = 1)/\pi$; this allows us to transform an $s(x)$ classifier into a $y(x)$ classifier (Bekker and Davis 2020). Another way to incorporate the class prior is to preprocess the data to transform it to a traditional supervised learning problem. Typically, this is done by introducing weights (Bekker and Davis 2020). Rebalancing methods aim to weight the examples so that the classifier trained on this data will give the same classification as the preprocessing method, while avoiding the problem of estimating the posterior $y(x)$ correctly (Elkan 2001; Hsieh, Natarajan, and Dhillon 2015; Lee and Liu 2003; Northcutt, Wu, and Chuang 2017); incorporating the label probabilities allows to count unlabeled examples partially as both positive and negative (Elkan and Noto 2008); while a large group of ERM-based methods aim to reweight the examples to be equivalent to a fully labeled dataset (Bekker, Robberechts, and Davis 2019; Steinberg and Cardell 1992; Kiryo et al. 2017; Hsieh, Natarajan, and Dhillon 2015; Plessis, Niu, and Sugiyama 2015; Plessis, Niu, and Sugiyama 2014). While many methods assume that the class prior π is known (Calvo, Larranaga, and Lozano 2007; Denis, Gilleron, and Letouzey 2005; Na et al. 2020), some methods tackle the problem of estimating it from the data (Bekker, Robberechts, and Davis 2019; Gong et al. 2021), frequently ignoring assumptions which are needed to ensure its identifiability. One of the weakest such conditions is an assumption that distribution of negative examples is not a convex mixture of distribution of positives and some probability distribution: $P_- = \alpha P_+ + (1 - \alpha)Q$ (Blanchard, Lee, and Scott 2010).

It is worth noting that many preexisting classification methods can be directly modified internally to work with PU data for case-control scenario with π known. Examples include decision trees (Denis, Gilleron, and Letouzey 2005), logistic regression (Ward et al. 2009), Naive Bayes (Denis, Laurent, and Tommasi 2003; Calvo, Larranaga, and Lozano 2007) and many others. The methods mentioned above are only a small subset of the existing literature on PU learning. For a more comprehensive overview, refer to (Bekker and Davis 2020).

Recently, the main focus of PU learning research has shifted towards deep learning. First methods utilizing neural networks were uPU and nnPU (Kiryo et al. 2017), with nnPU solving the issue of overfitting. Dist-PU (Zhao, Xu, et al. 2022), is an example of more recent deep learning method, approaching the PU problem from a label distribution perspective, i.e. building a classifier such that preparation of samples classified as positive is approximately π. Development of deep learning based methods led to the rise in no-SCAR PU research. It was mostly developed in single sample setting as nonconstant propensity score occurs naturally in this context. The first method proposed here was SAR-PU (Bekker, Robberechts, and Davis 2019) followed by LBE (Gong et al. 2021), utilizing expectation maximization (EM) algorithm to explicitly model both the posterior and propensity score; VAE-PU (Na et al. 2020), which uses variational autoencoders to generate examples from the missing positive unlabeled distribution; as well as double-logistic (Furmańczyk et al. 2023) and JERM (Rejchel, Teisseyre, and Mielniczuk 2024) methods, using the additional assumption of both posterior and propensity score being described by logistic functions. There were also attempts of introducing no-SCAR PU setting for case control data starting from Kato, Teshima, and Honda (2019), who proposed PUSB method. They are based on relaxing assumption that unlabeled data corresponds to original distribution of features and requires additional assumptions such as comonotonicity assumption (e.g. Assumption 1 in Kato, Teshima, and Honda (2019)). For recent attempts, we mention CoVPU (Liang et al. 2023) considering PU problem from an angle of positive distribution pollution, and PUDA (Tang, Pei, et al. 2022), focusing on knowledge graph completion task. Let us stress that in the thesis, apart from Chapter 3, no-SCAR setting for SS scenario is considered.

Recently, several papers discussing class imbalance in PU learning have appeared. Notably, Su, Chen, and Xu (2021) proposed ImbalancednnPU. This method is based on the nnPU (Kiryo et al. 2017) method, with part of its risk function reweighted to capture the class imbalance. Ortega Vázquez, Broucke, and De Weerdt (2023) proposed a decision tree based method utilizing the Hellinger distance as a imbalance insensitive splitting criterion. As handling imbalanced scenario is out of the scope of this thesis, we will not discuss these methods in detail – however, it is worth mentioning that this has become an active area of research recently.

Chapter 3

Comparison of PU learning scenarios under SCAR assumption

3.1 Problem introduction

At the core of every machine learning problem lies the data. The way the data is collected, the assumptions accounting for it, and the way it is structured are all crucial for the success of the learning process. Data acquisition process is even more crucial to understand in the context of PU learning, where understanding the how the data is collected lies at the core of the problem itself. Unfortunately, this topic is often overlooked in the literature, with many papers focusing on the algorithms themselves rather than the data they operate on. This chapter aims to fill this gap by providing a detailed comparison of two PU learning scenarios: single sample (SS) and case control (CC) under the SCAR assumption. We will show that the differences between the two scenarios are not only limited to the set sizes but also to the structure of the unlabeled data. We will also show that the structure of the unlabeled data has a significant impact on the performance of the algorithms designed for one scenario when applied to the other.

In a single sample scenario (SS; also called single-training-set or censoring scenario) we assume that there is some unknown distribution $P_{X,Y,S}$ such that $(X_i, Y_i, S_i), i = 1, \dots, n$ are independent and identically distributed (iid) random variables drawn from it. Observed data consists of $(X_i, S_i), i = 1, \dots, n$. In this chapter, in contrast to the rest of the thesis, we will assume that the positive observations are selected using SCAR assumption, i.e.

$$P(S = 1|Y = 1, X = x) = P(S = 1|Y = 1). \tag{3.1}$$

This in particular implies that the labeled examples are generated from the same distribution as the observations from the positive class i.e. $P_{X|S=1} = P_{X|Y=1}$, see Lemma 2.2 in Chapter 2.

In contrast, in case control scenario we observe two sets of examples, the first, called labeled class $L, X_1, \ldots, X_{n_1}$ pertaining to the positive class P and the second $X_{n_1+1}, \ldots, X_{n_1+n_2}$ drawn from a general population being the mixture of distributions $P_X = \pi P_{X|Y=1} + (1-\pi) P_{X|Y=-1}$ (unlabeled class U). Note that CC scenario is applicable when the example sets are drawn from two separate data bases; one pertaining to a general population and the other to the positive class, e.g. patients suffering from a certain disease. On the other hand, when e.g. a poll is conducted for a randomly chosen group of people who are asked whether they write text messages while driving ($Y = 1$) or not ($Y = -1$), this corresponds to SS scenario (with affirmative answer resulting in $S = 1$) – all of the data is collected at once, with no clear distinction between the sets, and the labeling is a result of purely probabilistic mechanism.

Note that although for CC case S would still denote a deterministic class indicator (labeled or unlabeled): $S = \mathbb{I}\{\text{observation belongs to } L\}$, it would be deterministic and not have probabilistic connotation as in SS case. We remark that the SCAR assumption is automatically satisfied in CC case if we consider the observations $X_1, \ldots, X_{n_1}$ as labeled: they are all generated from the distribution $P_{X|Y=1}$ and $X_{n_1+1}, \ldots, X_{n_1+n_2}$ are generated from P_X.

Observe that apart from minor differences concerning set sizes, which are random for single sample scenario and deterministic for case control, the main difference between the two scenarios lies in a structure of unlabeled set. In the case control scenario it corresponds to a general population which is a mixture of $P_{X|Y=1}$ and $P_{X|Y=-1}$ with mixing proportion $\pi = P(Y = 1)$. In contrast, for SS case it corresponds to the mixture with different mixing proportion

$$P_{X|S=0} = \frac{\pi - \pi c}{1 - \pi c} P_{X|Y=1} + \frac{1-\pi}{1-\pi c} P_{X|Y=-1}, \tag{3.2}$$

where $c = P(S = 1|Y = 1)$. This easily follows after noticing that out of proportion π of positive observations, proportion πc is labeled and $\pi - \pi c$ is unlabeled. In particular, we note that it follows from (3.2) that probability that positive element occurs among unlabeled data equals $\pi \times \frac{1-c}{1-\pi c}$ and is smaller or equal π being the probability of such occurrence in an original dataset. Thus, indeed, distributions of unlabeled examples differ in those cases. In particular note that for $c \approx 1$ unlabeled group in SS case consists mostly of negative observations in contrast to CC case when it corresponds to the original mixture. Figure 3.1 shows this behavior for two unit variance normal densities with means -2 and 2, respectively, $\pi = 0.5$ and $c = 0.1, 0.5$ and 0.9. Note that whereas for $c = 0.1$ the distributions between unlabeled data in both cases are almost indistinguishable, for $c = 0.9$ there is a striking difference between unlabeled distributions, the distribution being symmetric and bimodal in CC case whereas in SS case the second mode is barely discernible.

There is a legion of papers devoted to inference for PU data, and although they are mostly devoted to CC scenario (see the comprehensive review in Bekker and Davis (2020)), there are

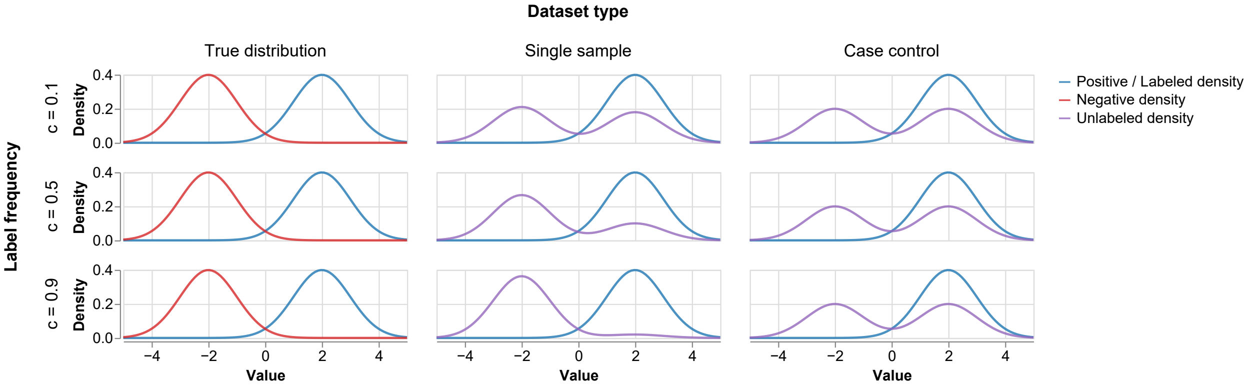


Figure 3.1: Comparison of labeled and unlabeled class density for SS and CC data

also many approaches which specifically deal with SS setup, starting from the seminal paper of Elkan and Noto (2008). However, it seems that understanding of the importance of sampling scenario for behavior of developed classifiers is limited and one can find many examples of careless use of CC-developed methods in SS scenario – sometimes accidentally, but often even after clearly stating the scenario assumptions. This also includes comparing performance of methods designed for a specific scenario with methods for the other scenario, which puts the latter at disadvantage. One good example in that case is a VAE-PU paper by Na et al. (2020), where the authors clearly state single sample scenario assumptions, but then proceed to evaluate a newly proposed method against five others, four of which work under case control assumption; while Confidence-Based PU learning paper by Tang, Xu, et al. (2025) demonstrates an opposite problem, applying single sample EN method (Płatek and Mielniczuk 2023) in an otherwise well-described case control framework. The purpose of this chapter is to show that this has important consequences and may lead to misleading conclusions especially when establishing a ranking of the classifiers with respect to some performance metrics. This is analyzed here for a particular case of Empirical Risk Minimizers, which play an important role in PU inference. In the following, as noted in the introduction we assume that the probability π of positive class is known.

3.2 Empirical Risk Minimization scenario differences

We give now the formal reason why application of ERM classifiers designed for one scenario will fail in the other scenario, save for very specific cases. We will focus on Empirical Risk Minimization (ERM) approach, consisting of finding minimizer of an empirical counterpart of the theoretical risk, as described in Section 2.3. Consider now a situation when it is applied for SS data using characteristics of the examples valid in CC case, namely that labeled examples pertain to $P_{X|Y=1}$ distribution and unlabeled ones are generated from P_X. Under SCAR the first assumption is valid as $P_{X|Y=1} = P_{X|S=1}$, whereas the second is not in view of (3.2). Consequently although the first terms in (2.8) and (2.10) are equal the second and the third terms in both expressions do not match, suggesting that Empirical Risk Minimization approach for case control situation can not be directly applied to single sample scenario and vice versa. Indeed, closer scrutiny of (2.8) and (2.10) yields the following fact.

Proposition 3.1. *(i) Applying formula (2.8) for SS scenario under assumption that $P_U = P_{X|S=0} = P_X$, is valid only when*

$$\mathbb{E}_{X|S=1}\, l(-g(X))\, \mathbb{E}_{X|S=0}\, l(-g(X)), \tag{3.3}$$

provided $P(S=1) > 0$.
(ii) Conversely, provided $P(S=1) > 0$, formula (2.10) for CC scenario under assumption that $P_U = P_{X|S=0} = P_X$, is valid only when

$$\mathbb{E}_{X|S=1}\, l(-g(X))\, \mathbb{E}_{X}\, l(-g(X)). \tag{3.4}$$

Proof. Application of (2.8) for SS scenario means that unlabeled population is treated as the original population, i.e. $P_U = P_{X|S=0} = P_X$ and thus (2.8) will take the form

$$\pi\, \mathbb{E}_{X|S=1}\, l(g(X)) + \mathbb{E}_{X|S=0}\, l(-g(X)) - \pi\, \mathbb{E}_{X|S=1}\, l(-g(X)) \tag{3.5}$$

as $\mathbb{E}_{X|Y=1} = \mathbb{E}_{X|S=1}$. This would be valid formula for $R(g)$ in SS scenario provided the above expression equals (2.10), which, taking into account again that under SCAR $P_{X|Y=1} = P_{X|S=1}$, yields

$$\begin{aligned} &\mathbb{E}_{X|S=0}\, l(-g(X)) - \pi\, \mathbb{E}_{X|S=1}\, l(-g(X)) \\ &= P(S=0)\, \mathbb{E}_{X|S=0}\, l(-g(X)) \\ &- P(Y=1, S=0)\, \mathbb{E}_{X|S=1}\, l(-g(X)). \end{aligned} \tag{3.6}$$

Thus, using $\pi - P(Y=1, S=0) = P(S=1)$, we have that the equality above is equivalent to

$$P(S=1)\, \mathbb{E}_{X|S=0}\, l(-g(X)) = P(S=1)\, \mathbb{E}_{X|S=1}\, l(-g(X)), \tag{3.7}$$

which yields the conclusion of (i).
Proof of (ii) is analogous. Formula (2.10) under assumption $P_U = P_{X|S=0} = P_X$ takes the form

$$\pi\, \mathbb{E}_{X|S=1}\, l(g(X)) + P(S=0)\, \mathbb{E}_X\, l(-g(X)) - P(Y=1, S=0)\, \mathbb{E}_{X|S=1}\, l(-g(X))$$

and equating it to (2.8) yields condition (3.4). □

Thus it follows that applying Empirical Risk Minimization (ERM) methods suitable for CC PU data (such as popular uPU and nnPU methods) to SS data directly, without modifying them, puts them at disadvantage.

Other methods derived for PU case control situation will also be affected when applied to single sample data. This is due to the fact that they necessarily use the information that unlabeled observations follow the general distribution. We have focused here on ERM methods as in this case it is possible to formally analyze the impact of scenario on the method; see Proposition 3.1. In the following we compare performance of nnPU$_{\mathrm{CC}}$, which optimizes non-negative modification of the empirical version of (2.8), with its counterpart adapted to SS data, which will be called nnPU$_{\mathrm{SS}}$ from now on. We compare their performance in a single sample case (when nnPU$_{\mathrm{SS}}$ should be used) and in a case control case (when the alternative method should be used).

Remark 3.2. *We note that both (3.3) and (3.4) are implied by $P_{X|S=1} = P_{X|S=0}$. However, it is easy to check that due to (3.2) and equality $P_{X|Y=1} = P_{X|S=1}$, this is equivalent to $P_{X|Y=1} = P_{X|Y=-1}$, which makes the original problem void, as distinguishing between classes is clearly impossible in this case.*

3.3 Scenario aware nnPU method

Consider first two popular estimators of (2.8) constructed for CC data. Note that $R(g)$ in Equation (2.8) can be split into three components: the first, corresponding to the labeled set risk (which we will denote as R^L in the algorithm), the second – to the risk with respect to the general distribution P_X (denoted by R^D), and finally – PU SCAR correction of the second term (R^{corr}). The first and the third component of Eq. (2.8) are independent of the scenario. The general distribution component $\mathbb{E}_X\, l(-g(X))$ can be consistently approximated by empirical average over U as observations in U are distributed according to P_X. Thus the direct plug-in estimator of (2.8) is (Plessis, Niu, and Sugiyama 2014)

$$\begin{aligned}\widehat{R}_{\mathrm{uPU_{CC}}}(g) = \frac{\pi}{n_L}\sum_{i:X_i\in L} l(g(X_i)) + \frac{1}{n_U}\sum_{i:X_i\in U} l(-g(X))\\ -\frac{\pi}{n_L}\sum_{i:X_i\in L} l(-g(X_i)).\end{aligned} \tag{3.8}$$

The minimizer of (3.8) is called unbiased PU estimator (uPU) and the index 'cc' is added in order to stress that it is derived for CC data. Its name is due to the fact that the empirical risk is unbiased: $\mathbb{E}\widehat{R}(g) = R(g)$. The nonnegative version (nnPU) is obtained when truncating the sum of the two last elements in (3.8) at 0. This is motivated by the fact that its theoretical counterpart $\mathbb{E}_X\, l(-g(X)) - \pi\,\mathbb{E}_{X|Y=1}\, l(-g(X))$ is nonnegative. Thus nnPU$_{\mathrm{CC}}$ estimator is minimizer of (see

Kiryo et al. (2017)):

$$\begin{aligned}\widehat{R}_{\mathrm{nnPU_{CC}}}(g) = {} & \frac{\pi}{n_L}\sum_{i:X_i\in L} l(g(X_i)) \\ & + \max\left(\frac{1}{n_U}\sum_{i:X_i\in U} l(-g(X_i)) - \frac{\pi}{n_L}\sum_{i:X_i\in L} l(-g(X_i)), 0\right),\end{aligned} \tag{3.9}$$

We consider now nnPU$_{\mathrm{SS}}$ estimator defined as the minimizer of the risk $R(g)$ for SS scenario based on (2.8). Note that for this scenario term $\mathbb{E}_X\, l(-g(X))$ can be approximated by empirical average over *all* observations disregarding their labels and thus we have direct plug-in estimator $\widehat{R}_{\mathrm{uPU_{SS}}}$

$$\begin{aligned}\widehat{R}_{\mathrm{uPU_{SS}}} = {} & \frac{\pi}{n_L}\sum_{X_i\in L} l(g(X_i)) \\ & + \frac{1}{n}\sum_{X_i\in L\cup U} l(-g(X_i)) - \frac{\pi}{n_L}\sum_{x\in L} l(-g(X_i))\end{aligned} \tag{3.10}$$

and its nonnegative version $\widehat{R}_{\mathrm{nnPU_{SS}}}$:

$$\begin{aligned}\widehat{R}_{\mathrm{nnPU_{SS}}} = {} & \frac{\pi}{n_L}\sum_{X_i\in L} l(g(X_i)) \\ & + \max\left(\frac{1}{n}\sum_{X_i\in L\cup U} l(-g(X_i)) - \frac{\pi}{n_{L_i}}\sum_{x\in L} l(-g(X_i)), 0\right).\end{aligned} \tag{3.11}$$

Similarly to the case control case, $\widehat{R}_{\mathrm{uPU_{SS}}}$ is also unbiased estimator of $R(g)$, as for this property it is immaterial that the summands in Equation (3.10) are dependent. The final, labeling-scenario-aware training procedure is described in detail in Algorithm 1. Note that the algorithm incorporates the change of the gradient sign when truncation occurs as advocated in the original nnPU algorithm (Kato, Teshima, and Honda 2019). We also note that in order to obtain the version of nnPU$_{\mathrm{SS}}$ algorithm it is only necessary to change in the definition of $\widehat{R}_{\mathrm{uPU_{SS}}}$, the value of R^D to the average calculated over all data and not over U set.

Remark 3.3. *We also note that it is possible to obtain $\widehat{R}_{uPU_{SS}}$ based on (2.10). Namely plug-in version of (2.10) has the form*

$$\begin{aligned}& \frac{\pi}{n_L}\sum_{i:X_i\in L} l(g(X_i)) + \frac{1}{n}\sum_{i:X_i\in U} l(-g(X)) \\ & -\left(\pi - \frac{n_L}{n}\right)\frac{1}{n}\sum_{i:X_i\in L} l(-g(X_i)),\end{aligned} \tag{3.12}$$

where $n_L = |\{i : S_i = 1\}|$ and $n_U = n - n_L$ and $P(Y = 1, S = 0)$ is estimated by $\pi - \widehat{P}(S = 1) = \pi - \frac{n_L}{n}$. Its non-negative version is defined analogously. However, by splitting the sum $\frac{1}{n}\sum_{X_i \in L \cup U} l(-g(X_i))$ into the sums over L and U it is easy to see that (3.12) equals (3.10). The form of the risk function in (3.10), while less useful from the theoretical point of view, has an important advantage: namely that only a single component differs between (3.8) and (3.10). This allows for easier comparison of those risks over the course of our experiments and is thus used in our implementation of the nnPU$_{SS}$ algorithm.

We discuss the reason the negative part of $\widehat{R}_{\mathrm{uPU}_{\mathrm{CC}}}$ is biased downwards when applied to single sample data. The first and the third terms in (3.8) are consistent estimators of their theoretical counterparts, the problem occurs for the second term. Namely, large contribution to the sum $\sum_{i \in U} l(-g(X_i))$ corresponds to positive elements which are likely to be assigned to positive class by classification function $g(x)$ but will be assigned to negative class by $-g(x)$ thus producing large loss. However, when uPU$_{\mathrm{CC}}$ is applied to SS data the proportion of positive elements among unlabeled data is smaller than in the general population as discussed in the introduction. This results in downward bias of the negative part of ERM. Thus truncation at 0 is more likely to occur here than for CC scenario. Conversely, when nnPU$_{\mathrm{SS}}$ is applied for CC data the term $n^{-1}\sum_{i:S_i=0} l(-g(X_i))$ in (3.12) is larger than for SS data, as the proportion of positive observations among unlabeled observations is larger. Note that the both cases of misuse are not entirely symmetric as erroneous application of $\widehat{R}_{\mathrm{nnPU}_{\mathrm{CC}}}(g)$ to SS data results in more likely truncation, whereas for $\widehat{R}_{\mathrm{nnPU}_{\mathrm{SS}}}(g)$ used for CC data truncation is less likely.

3.4 Scenario-fair dataset creation

Due to the fact that the structure of the unlabeled data is different in both scenarios, creating the datasets which ensure equal conditions for evaluation of both scenarios is not straightforward. At one hand, it requires to ensure that the expected fraction of labeled observations for a given c is the same in both scenarios. On the other hand, the sizes of the datasets should be equal. As this is an issue which was not solved in the literature before, we show how to achieve this in the following.

Consider the problem of creating a scenario-fair case control dataset. In the following n will stand for the total number of observations. In CC scenario c will mean a ratio of labeled observations ($S = 1$) to all positive ones ($Y = 1$). In order to ensure that the expected fraction of labeled observations for a given c is the same in both scenarios we pick $c \times \pi \times n$ observations from positive class and $(1-c) \times n$ from the whole data set. Thus $c \times \pi \times n + \pi \times (1-c) \times n = \pi \times n$ is an expected number of observations from the positive class in the dataset and fraction c of them will be labeled on average. In order to ensure that the chosen dataset has size equal to n,

Algorithm 1: Scenario-aware nnPU algorithm

```
Input: Positive-unlabeled dataset $X = (L, U)$, $\pi$ – class prior, $n$ – number of training
       items, hyperparameters $\beta$ and $\gamma$ (as described in detail in Kato, Teshima, and
       Honda (2019)).
repeat
    Split $X$ into $k$ minibatches
    forall minibatch $M_i = (L_i, U_i)$ in $X$ do
        Calculate labeled risk component $R^L$
            $R^L = \pi \frac{1}{n_{L_i}} \sum_{x \in L_i} l(g(x)),$
        if $nnPU_{SS}$ then
            Calculate general distribution component $R^D$ based on the whole dataset
                $R^D = R^D_{\mathrm{SS}} = \frac{1}{n} \sum_{x \in L_i \cup U_i} l(-g(x)),$
        else if $nnPU_{CC}$ then
            Calculate general distribution component $R^D$ based on the unlabeled set
                $R^D = R^D_{\mathrm{CC}} = \frac{1}{n_{U_i}} \sum_{x \in U_i} l(-g(x)),$
        Calculate PU SCAR correction $R^{\mathrm{corr}}$
            $R^{\mathrm{corr}} = \pi \frac{1}{n_{L_i}} \sum_{x \in L_i} l(-g(x)),$
        if nonnegative risk component $R^D - R^{corr} \leq -\beta$ then
            Perform gradient descent for unbiased risk $R = R^L + (R^D - R^{\mathrm{corr}})$ with step size
              $\eta$.
        else
            Update model parameters using surrogate $R^{\mathrm{surr}} = R^{\mathrm{corr}} - R^D$ with discounted
              step size $\gamma\eta$.
        end
    end
until not converged;
```

both sizes should be increased $A = (1 - c(1 - \pi))^{-1}$ times i.e. size of chosen labeled dataset should be $A \times c \times \pi \times n$ and $A \times (1 - c) \times n$ for the unlabeled one. Note that both sets are not necessarily disjoint but this should not affect the performance of the rule as empirical risk function is an unbiased estimator of theoretical risk also in this case.

We stress that examination of these two sampling scenarios shows that the meaning of some parameters is different in both methods, e.g. for single sample scenario $c = P(S = 1|Y = 1) = 1$ means that every positive observation will be labeled and thus the set $(X_i, S_i)_{i=1}^n$ is generated from the distribution $P_{X,Y}$. In case control scenario it means that every observation for positive class is sampled, however unlabeled set will be empty as otherwise (see the calculation above) c will be strictly smaller than 1.

3.5 Experiments

We consider two sampling scenarios:

(a) **Single sample scenario**. For a given data set for which π is taken as a fraction of positive class in it, we sample $n = 1000$ elements randomly without replacement and label them using SCAR scenario with varying $c = P(S = 1|Y = 1)$. We apply SS and CC methods to the obtained dataset.

(b) **Case control scenario.** We adapt the labeling procedure described in Section 3.4 to ensure fairness of the experiments.

We performed experiments on a collection of 18 diverse datasets from different domains: image, text and tabular classification. Dataset details are shown in Table 3.1 – the test ensemble contains datasets of various sizes and class balances. In order to focus on risk function differences rather than on tuning particular network architectures, as well as to shorten training time, we used pretrained embeddings for text and image data (`all-MiniLM-L6-v2` (Wang, Wei, et al. 2020) and `swiftformer-xs` (Shaker et al. 2023) respectively). For processed data classification, we used 5-layer feed-forward neural network, matching the one used in Kato, Teshima, and Honda (2019). Similarly to this paper, we kept the default values of hyper-parameters: $\beta = 0$ and $\gamma = 1$ (see Algorithm 1 for their description). Based on the each dataset, we synthetically created both single sample and case control problem, as described above. We either use a pre-existing train-test split or, in cases where it is not available, split dataset with a 80-20 training-test ratio.

During testing, multiple label frequency c levels, ranging from 0.1 to 0.9, were applied. Each experiment (for a given dataset, scenario, label frequency and method combination) was repeated 10 times with a different random seed. As the chapter's aim is to emphasize impact

Table 3.1: Dataset statistics

Dataset	Data type	Samples	Features	π
CIFAR	Image	50000	10780	0.60
MNIST	Image	60000	10780	0.51
FashionMNIST	Image	60000	10780	0.50
EuroSAT	Image	21600	10780	0.30
Chest X-ray	Image	4077	10780	0.73
Snacks	Image	4838	10780	0.41
DogFood	Image	2250	10780	0.33
Beans	Image	1034	10780	0.33
Oxford Pets	Image	5912	10780	0.33
20News	Text	11314	384	0.56
IMDB	Text	25000	384	0.50
HateSpeech	Text	8561	384	0.11
SMSSpam	Text	4459	384	0.13
PoemSentiment	Text	892	384	0.15
Credit	Tabular	13371	10	0.50
California	Tabular	16507	8	0.50
Wine	Tabular	2043	11	0.51
Electricity	Tabular	30779	7	0.50

of correct method selection, we focused on comparing nnPU$_{\text{SS}}$ and nnPU$_{\text{CC}}$ methods and did not include comparisons with any external classifiers. Implementation of Algorithm 1 and all experiments' code are publicly available on GitHub[1].

3.6 Results

Experiment results are summarized in Tables 3.2 and 3.3, for SS and CC data respectively (results for metrics other than accuracy: precision, recall and F1 score can be found in the GitHub repository; conclusions for F1 score largely correlate with the accuracy-based analysis below). The key observation is that in both cases the advantage of the correctly specified method (defined as the difference of respective accuracies and denoted by Δ) starts for c as low $c = 0.1$, but as the proportion of labeled examples in the dataset increases it tends to significantly increase. Upon closer inspection, this is expected – as apparent in Figure 3.1, the difference in unlabeled set structure is subtle when label frequency is low, but becomes very apparent when c increases. That causes both methods' performance to be close when label frequency is low, but for high c values, when scenario dependency deepens, they start to diverge.

Due to the reasons above, it is worth inspecting results for $c = 0.9$ in detail, as this is when scenario differences are the most distinctive. In the vast majority of cases, the advantage of correctly specified method is very significant – this is most apparent for some image datasets, such as MNIST and CIFAR. Rare cases where performance of both methods is similar are more

[1] `https://github.com/wawrzenczyka/nnPUss`

Table 3.2: Test accuracy, single sample datasets. Δ indicates accuracy difference between scenario-appropriate nnPU$_{\text{SS}}$ method and ill-specified nnPU$_{\text{CC}}$ method.

c	Model	Beans	CIFAR	Chest X-ray	DogFood	EuroSAT	FashionMNIST	MNIST	Oxford Pets	Snacks
0.1	nnPUcc	81.88	92.69	88.93	87.69	90.26	97.16	95.16	86.80	74.96
	nnPUss	79.22	91.53	89.62	86.15	87.77	95.14	93.71	83.50	75.05
	Δ	-2.66	-1.16	0.69	-1.55	-2.49	-2.02	-1.45	-3.29	0.09
0.3	nnPUcc	89.92	92.01	91.49	97.56	94.19	96.68	95.23	95.86	80.68
	nnPUss	89.06	93.80	92.71	95.21	91.71	96.96	96.77	89.76	81.34
	Δ	-0.86	1.78	1.22	-2.35	-2.48	0.28	1.53	-6.10	0.66
0.5	nnPUcc	91.09	87.52	91.25	98.77	93.56	94.86	92.41	98.12	81.83
	nnPUss	91.80	95.04	93.40	98.13	94.21	97.98	98.23	93.50	84.96
	Δ	0.70	7.52	2.15	-0.64	0.65	3.12	5.82	-4.62	3.13
0.7	nnPUcc	91.64	80.18	88.83	98.84	90.62	82.96	85.41	97.50	81.63
	nnPUss	94.22	96.46	94.05	99.33	95.67	99.10	98.98	96.20	87.25
	Δ	2.58	16.27	5.22	0.49	5.06	16.14	13.57	-1.30	5.62
0.9	nnPUcc	91.33	75.28	85.09	98.61	87.60	68.54	67.81	96.39	81.39
	nnPUss	95.70	97.57	95.15	99.80	96.94	99.43	99.19	98.95	89.39
	Δ	4.37	22.29	10.07	1.19	9.34	30.89	31.38	2.55	8.00

c	Model	California	Credit	Electricity	Wine	20News	HateSpeech	IMDB	PoemSentiment	SMSSpam
0.1	nnPUcc	81.71	64.48	75.16	69.22	79.95	88.94	73.31	84.71	88.70
	nnPUss	81.77	64.48	74.93	68.51	78.61	88.94	72.79	84.81	89.39
	Δ	0.06	0.01	-0.23	-0.70	-1.34	0.00	-0.52	0.10	0.69
0.3	nnPUcc	83.20	66.59	78.42	73.89	82.50	89.52	77.00	85.00	96.55
	nnPUss	84.09	67.85	78.00	74.03	81.38	89.41	75.68	85.87	95.62
	Δ	0.89	1.26	-0.42	0.14	-1.12	-0.10	-1.32	0.87	-0.92
0.5	nnPUcc	82.46	63.99	79.54	75.95	81.87	89.29	78.00	86.15	97.87
	nnPUss	85.31	66.23	79.64	75.97	83.28	89.32	76.83	87.60	97.18
	Δ	2.85	2.24	0.11	0.02	1.40	0.03	-1.16	1.44	-0.68
0.7	nnPUcc	80.13	62.18	78.55	76.18	78.99	89.01	77.13	89.04	98.48
	nnPUss	86.27	63.56	80.67	78.16	84.43	89.51	78.27	89.71	98.15
	Δ	6.14	1.37	2.12	1.98	5.44	0.50	1.15	0.67	-0.32
0.9	nnPUcc	76.82	61.33	75.87	74.72	74.27	88.82	74.76	90.00	98.72
	nnPUss	86.36	60.54	81.20	79.47	85.75	89.65	79.34	90.87	98.33
	Δ	9.54	-0.79	5.33	4.76	11.48	0.83	4.58	0.87	-0.39

Table 3.3: Test accuracy, case control datasets. Δ indicates accuracy difference between scenario-appropriate nnPU_{CC} method and ill-specified nnPU_{SS} method.

c	Model	Beans	CIFAR	Chest X-ray	DogFood	EuroSAT	Fashion MNIST	MNIST	Oxford Pets	Snacks
0.1	nnPUss	77.91	93.06	89.60	85.75	86.30	94.68	93.36	82.98	73.10
	nnPUcc	78.37	93.52	88.83	86.92	89.39	96.93	95.49	86.30	72.67
	Δ	0.47	0.46	-0.77	1.16	3.09	2.25	2.13	3.31	-0.43
0.3	nnPUss	83.88	94.40	92.99	92.43	88.26	94.00	93.50	84.24	80.40
	nnPUcc	83.64	95.48	92.15	96.44	93.26	98.47	97.56	92.56	80.77
	Δ	-0.23	1.08	-0.84	4.01	5.00	4.47	4.06	8.32	0.37
0.5	nnPUss	85.81	89.25	92.58	93.46	84.07	85.91	83.70	84.98	74.28
	nnPUcc	85.66	96.75	93.20	97.50	94.54	99.12	98.58	95.16	83.04
	Δ	-0.16	7.50	0.62	4.05	10.47	13.21	14.88	10.19	8.76
0.7	nnPUss	81.94	81.83	82.27	88.26	76.42	83.24	85.71	79.87	69.13
	nnPUcc	85.04	97.74	90.38	96.14	95.31	99.37	99.05	96.38	80.30
	Δ	3.10	15.91	8.11	7.88	18.89	16.13	13.33	16.51	11.18
0.9	nnPUss	44.34	89.43	85.44	30.24	24.56	87.21	86.61	25.70	46.79
	nnPUcc	62.40	98.71	79.86	74.77	95.55	99.58	99.25	91.36	68.05
	Δ	18.06	9.28	-5.57	44.53	70.99	12.37	12.64	65.67	21.26

c	Model	California	Credit	Electricity	Wine	20News	HateSpeech	IMDB	PoemSentiment	SMSSpam
0.1	nnPUss	81.27	63.93	74.60	68.42	78.46	88.00	72.55	82.57	89.27
	nnPUcc	81.27	64.07	74.82	69.61	79.82	88.00	73.14	82.76	88.68
	Δ	0.00	0.14	0.22	1.19	1.36	0.00	0.59	0.19	-0.58
0.3	nnPUss	83.74	67.59	76.33	69.96	79.63	86.40	71.35	82.86	94.20
	nnPUcc	84.60	67.89	76.95	73.55	83.53	86.44	75.72	82.86	95.64
	Δ	0.86	0.30	0.62	3.59	3.90	0.05	4.36	0.00	1.43
0.5	nnPUss	84.03	69.67	77.08	72.15	73.08	83.00	71.01	82.10	92.26
	nnPUcc	85.78	70.68	77.51	76.46	85.86	84.28	76.32	82.38	96.83
	Δ	1.75	1.01	0.43	4.32	12.78	1.28	5.31	0.29	4.57
0.7	nnPUss	85.45	71.69	77.47	72.11	72.80	71.71	70.83	72.38	77.40
	nnPUcc	87.44	73.94	78.20	78.63	88.64	80.01	77.38	79.90	96.48
	Δ	1.99	2.25	0.73	6.52	15.84	8.31	6.55	7.52	19.08
0.9	nnPUss	83.13	71.34	76.06	70.78	81.35	44.69	70.96	34.57	40.71
	nnPUcc	89.85	76.13	79.58	82.97	92.80	63.64	79.70	59.14	89.85
	Δ	6.72	4.79	3.51	12.19	11.46	18.95	8.74	24.57	49.14

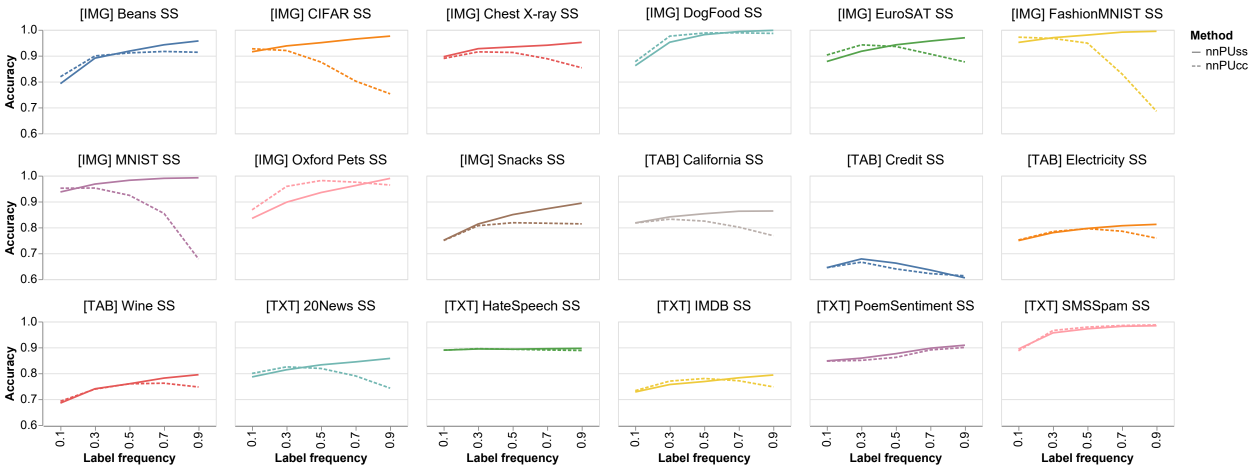


Figure 3.2: Change of accuracy with label frequency increase for single sample datasets

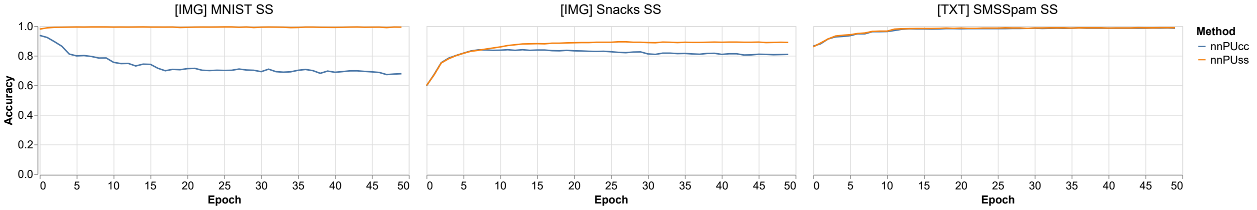


Figure 3.3: Test accuracy per epoch, selected single sample datasets, $c = 0.9$

common for text and tabular data. Note that the for the correct method, in vast majority of cases accuracy tends to steadily increase as the label frequency rises, while the alternative methods' performance starts to drop off. Figure 3.2 illustrates this phenomenon for SS datasets, but the former statement holds true for case control data as well. Depicted behavior seems to indicate overfitting of the ill-specified method for high c values, which, as apparent in Figure 3.3, occurs on multiple datasets and is the major cause of the classification performance deterioration. We stress that as in the positive-unlabeled problems we have no access to the fully labeled validation dataset, robustness to overfitting is drastically more important for any PU learning method than in the binary classification task. Overall, ill-specified methods are prone to overfitting after only a couple of epochs. Rare cases, where they do not overfit, match cases, where the overall accuracy of both methods stay close.

In order to understand the behavior of both methods better, closer inspection of a change of risk values during training is crucial. Figure 3.4 illustrates a typical risk component changes throughout the learning process for both nnPU_{SS} and nnPU_{CC}. Note that even when wrong method is applied (nnPU_{CC} for single sample datasets and vice versa) the "correct" risk values still decrease throughout training. This suggests that for both nnPU_{SS} and nnPU_{CC} gradients might point in the similar directions, and explains why (especially in the initial training epochs) learning does not fail completely. It is also interesting to consider reasons why this does not

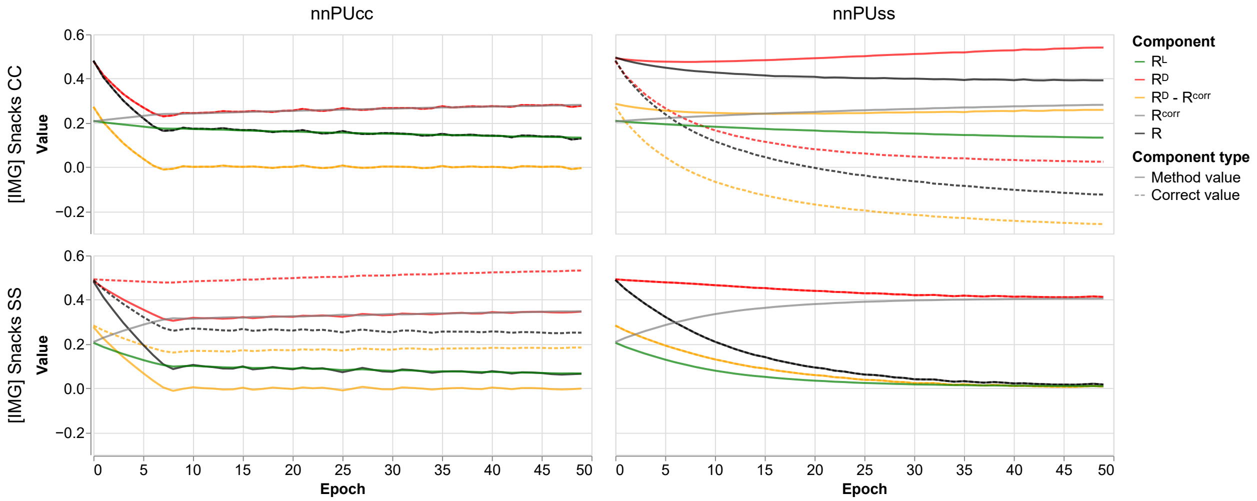


Figure 3.4: Risk components per epoch, Snacks dataset, $c = 0.9$. "Method" values refer to risk values obtained during training, whereas "Correct" values – to the ones which would be obtained in the given epoch if scenario-aware risk would be applied.

hold true throughout whole training. When attempting to train nnPU$_{\text{SS}}$ on the case control data (Fig. 3.4, upper right), note that for the correct, case control-tailored risk value, non-negative risk component $R^D - R^{\text{corr}}$ drops well below 0 – which is strongly undesirable and is normally discouraged in Algorithm 1, step 11. Due to this, severe overfitting occurs only a few epochs into training. In the symmetric case, nnPU$_{\text{CC}}$ on the single sample data (Fig. 3.4, lower left), the problem is slightly different. Due to R^D being underestimated by the algorithm (as it expect more positive examples in the unlabeled dataset), nonnegative component of the risk quickly falls towards 0 and starts oscillating in its vicinity. This time step 11 of Algorithm 1 is activated very early. However, by the view of the correct, scenario-aware risk function, these updates are counterproductive – they are only valid when nonnegative component stops being positive – and counteract quasi-valid updates of the model, causing performance decline.

3.7 Conclusions

In this chapter, we discussed issues with automatic application of PU learning methods without taking into account labeling scenario constraints. To this end, we introduced nnPU$_{\text{SS}}$ – equivalent of nnPU (denoted in this chapter as nnPU$_{\text{CC}}$), a popular benchmark method in PU learning papers, tailored to SS scenario – and through experiments, we exposed dangers of incorrect method selection, and identified reasons for such behavior. Obtained results indicate that for high-enough label frequencies (approximately $c \geq 0.5$), using algorithms not devised for a particular scenario might incur a major performance loss, and – especially in the case of ranking classifiers according to their performance – might put the method investigated at the huge disadvantage. We also introduced a correct form of nnPU$_{\text{SS}}$ classifier for SS data and indicated that its algorithm can be easily obtained from the algorithm of nnPU$_{\text{SS}}$ classifier by changing one line of its code.

Chapter 4

No-SCAR PU learning problem

4.1 Problem introduction

In the previous chapter, we have discussed the SCAR PU learning problem, where the labeled data distribution matches the distribution of the positive class. This property greatly reduces the problem complexity; this was evident in the nnPU case, as the SCAR risk function was simplified by applying the SCAR assumption. However, in many real-world applications, this assumption is not valid – in practice, it is much more common that labeling introduces some sort of bias, be it age or weight in medical risk analyses, ease of labeling in expert reliant fields, or any other factor which affects the uniformity of the labeling process.

In this chapter, we will discuss the no-SCAR PU learning problem, where the labeled data distribution does not match the distribution of the positive class. This makes the problem significantly more complex, and not solvable without introducing additional assumptions. Accounting for bias in labeling is an important research trend in recent years to combat limitations of the SCAR approach applied in the earlier works. This lead to proposing several new methods in the recent years: most prominent of those are the LBE (Gong et al. 2021) method modeling propensity score explicitly using an approach based on Lemma 2.1, and SAR-PU (Bekker and Davis 2018b), using an EM procedure to iteratively improve posterior probability predictions. This chapter will center around yet another no-SCAR method, called VAE-PU, which uses an variational autoencoder to generate positive unlabeled examples and thus avoiding the need to model the propensity. We first introduce the model predecessors and explain the basis of its operations; after that, we will discuss VAE-PU's issues, and propose a series of improvements and key modifications which led to creation of the VAE-PU+OCC model. We describe the model and comparing its performance on several datasets, contrasting its performance with the base VAE-PU, as well as LBE and SAR-PU.

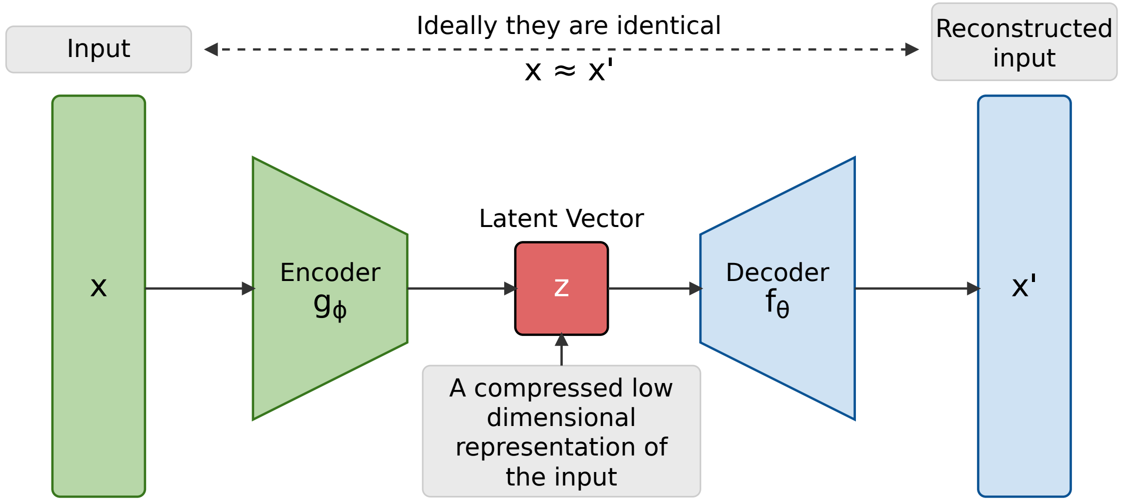

Input
Ideally they are identical
x ≈ x'
Reconstructed input
x
Encoder
g_ϕ
Latent Vector
z
A compressed low dimensional representation of the input
Decoder
f_θ
x'

Figure 4.1: Simple autoencoder architecture

4.2 Variational autoencoders

Autoencoders (**AE**) (see e.g. Bank, Koenigstein, and Giryes (2023)) are a type of neural network architecture that aims to reproduce its input as an output. They work by compressing the input set x into a latent-space representation using an **encoder**, and then translating that representation back into input space reconstruction $\widehat{x}$ via a **decoder**:

$$\begin{aligned} z &= \text{Encoder}(x) \\ \widehat{x} &= \text{Decoder}(z) \end{aligned} \tag{4.1}$$

The latent space is a lower-dimensional representation of the input data, which forces the model to learn the key properties allowing for precise example reconstruction. Figure 4.1 shows a graphical representation of the simple autoencoder architecture described above. The autoencoder is trained to minimize the **reconstruction error**, that is the difference between the input and the output, which is typically measured using the **mean squared error** (**MSE**) loss function:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(\widehat{x}_i - x_i)^2 \tag{4.2}$$

The latent space of an autoencoder is unregularized, meaning that it has not been explicitly structured or constrained during training. This lack of imposed structure has important consequences. First, the representation in the latent space may not be a continuous function of the input, meaning that small changes in the input data may result in large changes in the latent representation. The latent space may also not be smooth, meaning that nearby points in the input space may not be nearby in the latent space. Another issue is that some of the regions of the latent space may not contain any observations, making it difficult to generate new data points from these regions. That means that attempting to decode the latent space sample from outside the training data distribution may result in a nonsensical output. This can make it difficult to generate new observations from the latent space, as the model may not be able to smoothly interpolate between known data points. Due to all of the above limitations, applications of pure autoencoders are fairly limited, mostly to the tasks like data compression and denoising (Bank, Koenigstein, and Giryes 2023; Kögler et al. 2024; Lee, Ozger, et al. 2021).

To address those issues, **variational autoencoders** (**VAE**) (Kingma and Welling 2019) introduce a regularization term that forces the latent space to follow a specific distribution. This is achieved by adding a **Kullback-Leibler** (**KL**) divergence term to the reconstruction term, which measures the difference between the conditional latent space distribution and a prior distribution. Also, the reconstruction term itself is modified (see Eq. (4.15) below).

The KL divergence term between two distributions P and Q on the same support is defined as (see e.g. Cover and Thomas (1991)):

$$D_{\mathrm{KL}}(P||Q) = \sum_{x} P(x) \log \frac{P(x)}{Q(x)} \tag{4.3}$$

Any type of distribution can be used as a latent distribution, but the most common choice prevalent in literature is multivariate standard normal distribution.

The other core property of VAEs is enabling probabilistic generation of new examples. Instead of directly encoding the input data into the latent space, VAEs encode the input data into the parameters of latent probability distribution:

$$[\mu; \sigma] = \mathrm{Encoder}(x) \tag{4.4}$$

The latent space observations can be then sampled from this distribution (for simplicity, we will assume conditional latent distribution $q(z|x)$ to be a multivariate normal distribution $\mathcal{N}(\mu, \Sigma)$, where $\mu = (\mu_1, \mu_2, \ldots, \mu_J)$ and $\Sigma = \mathrm{diag}(\sigma_1^2, \sigma_2^2, \ldots, \sigma_J^2)$):

$$q(z|x) = \mathcal{N}(\mu, \Sigma) \tag{4.5}$$

The decoder then generates the output based on the sampled latent space, exactly as in the simple autoencoder case:

$$\widehat{x} = \mathrm{Decoder}(z) \tag{4.6}$$

The VAE architecture described above is shown in Figure 4.2.

The formulas used in the description above suffer from a key problem – using Equation (4.5) directly is impossible, as it cannot be optimized using backpropagation algorithm. To address

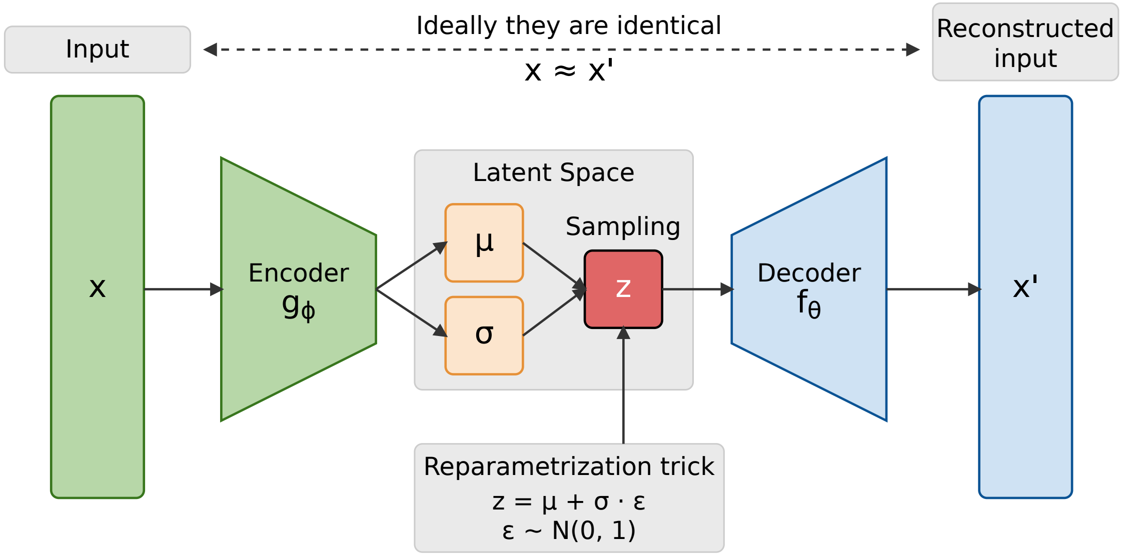


Figure 4.2: Variational autoencoder architecture

this issue, the **reparametrization trick** (Kingma and Welling 2019) is used. Instead of sampling from the distribution directly, the latent space observation is sampled from a standard normal distribution and then transformed into the desired distribution using the mean and standard deviation obtained from the encoder:

$$z = \mu + \sigma \cdot \epsilon \tag{4.7}$$

where ϵ is a sample from the standard normal distribution $\epsilon \sim \mathcal{N}(0,1)$. This can be done as due to its properties, z in Equation (4.7) has $\mathcal{N}(\mu, \sigma^2)$ distribution. This way, the randomness source ϵ is decoupled from learnable model parameters μ and σ, allowing for the use of backpropagation while still preserving the stochastic nature of the model. Note that reparametrization trick is applied to every component of z separately, that is the above notation is a shorthand for $z_i = \mu_i + \sigma_i \cdot \epsilon_i, i = 1, ..., J$, which is feasible due to assumption that the components of z are independent. The final reconstruction process can be then summarized as follows:

$$\begin{aligned} [\mu, \sigma] &= \text{Encoder}(x) \\ z &= \mu + \sigma \cdot \epsilon \\ \widehat{x} &= \text{Decoder}(z) \end{aligned} \tag{4.8}$$

In contrast to the simple autoencoder, the VAE architecture allows for powerful generative capabilities. Structured and regularized nature of the latent space allows for probabilistic generation of new examples and smooth interpolation between known observations, and the reparametrization trick enables the use of backpropagation for optimization. This makes VAEs a powerful tool for a wide range of tasks, including image and text generation (Wang, Gan, et al. 2019), representation learning (Sadok et al. 2024), and anomaly detection (Nguyen et al. 2024) in addition to capabilities offered by the autoencoder models (Kingma and Welling 2019; Jiang et al. 2017).

The core objective of training a VAE is to maximize the likelihood of the observed data $p(x)$. However, directly maximizing this likelihood is infeasible due to representation of $p(x)$ as the integral over the latent variable z:

$$p(x) = \int p(x,z)\,\mathrm{d}z \tag{4.9}$$

To address this issue, we approximate true posterior $p(z|x)$ by the variational approximation $q(z|x)$. The goal is to make $q(z|x)$ as close as possible to $p(z|x)$. This can be achieved by maximizing the **Evidence Lower Bound** (**ELBO**), which is a lower bound on the log-likelihood of the observed data.

The ELBO can be derived as follows. First, we start with the log-likelihood of the observed data:

$$\log p(x) = \log \int p(x,z)\,\mathrm{d}z \tag{4.10}$$

We then introduce the variational approximation $q(z|x)$ inside the logarithm:

$$\log p(x) = \log \int q(z|x)\frac{p(x,z)}{q(z|x)}\,\mathrm{d}z = \log \mathbb{E}_{q(z|x)}\left[\frac{p(x,z)}{q(z|x)}\right] \tag{4.11}$$

Now, we will use Jensen's inequality, which states that for any convex function ϕ, $\mathbb{E}[\phi(x)] \geq \phi(\mathbb{E}[x])$. In our case, we will use the logarithm function, which is concave, so we will reverse the inequality – we can move the logarithm inside the expectation to obtain a lower bound:

$$\log p(x) \geq \mathbb{E}_{q(z|x)}\left[\log \frac{p(x,z)}{q(z|x)}\right] \tag{4.12}$$

The right-hand side of the inequality is the ELBO, which we denote as $\mathscr{L}_{\text{ELBO}}(x)$:

$$\begin{aligned}\mathscr{L}_{\text{ELBO}}(x) &= \mathbb{E}_{q(z|x)}\left[\log \frac{p(x,z)}{q(z|x)}\right] \\ &= \mathbb{E}_{q(z|x)}\left[\log \frac{p(x|z)p(z)}{q(z|x)}\right]\end{aligned} \tag{4.13}$$

This can be rewritten as:

$$
\begin{aligned}
\mathscr{L}_{\text{ELBO}}(x) &= \mathbb{E}_{q(z|x)}\left[\log p(x|z)\right] + \mathbb{E}_{q(z|x)}\left[\log \frac{p(z)}{q(z|x)}\right] \\
&= \mathbb{E}_{q(z|x)}\left[\log p(x|z)\right] + \int q(z|x) \log \frac{p(z)}{q(z|x)}\, \mathrm{d}z \\
&= \mathbb{E}_{q(z|x)}\left[\log p(x|z)\right] - \int q(z|x) \log \frac{q(z|x)}{p(z)}\, \mathrm{d}z
\end{aligned}
\tag{4.14}
$$

The second term is the KL divergence between the approximate posterior $q(z|x)$ and the prior $p(z)$, thus the ELBO can be written as:

$$
\mathscr{L}_{\text{ELBO}}(x) = \underbrace{\mathbb{E}_{q(z|x)}[\log p(x|z)]}_{\text{reconstruction term}} - \underbrace{D_{\text{KL}}\left(q(z|x)||p(z)\right)}_{\text{regularization term}}
\tag{4.15}
$$

Maximizing the ELBO with respect to the parameters of the encoder and decoder networks is equivalent to maximizing the lower bound of log-likelihood of the observed data (which improves the reconstruction capabilities of the network), while simultaneously minimizing (due to negative sign) the KL divergence between the approximate posterior and the prior, serving as a regularization term for the latent space. Thus, optimization of ELBO ensures that the latent space is structured and regularized, enabling the generative capabilities of VAE via the interplay of reconstruction and regularization terms.

4.3 Variational Deep Embedding

VAE-PU model, core to the remainder of this thesis, is rooted in another variational model – **Variational Deep Embedding** (**VaDE**) (Jiang et al. 2017). VaDE is a generative model that combines the power of VAEs with the clustering capabilities of **Gaussian Mixture Models** (**GMMs**). The model is designed to learn a deep representation of the data that is both structured and regularized (close to the latent prior), enabling clustering of the observations coupled with powerful generative capabilities, making generation of examples belonging to any of the specified clusters feasible. VaDE can be viewed as a generalization of VAE, in a sense that a single Gaussian distribution is replaced by a mixture of Gaussians, allowing for more complex composition of the latent space and making it more suitable for clustering tasks.

The Gaussian Mixture Model is a probabilistic model that assumes that the data is generated by a mixture of several Gaussian distributions. The model is defined by the number of components K, the means μ_k, variances σ_k^2, and mixing coefficients π_k for each component k. The probability

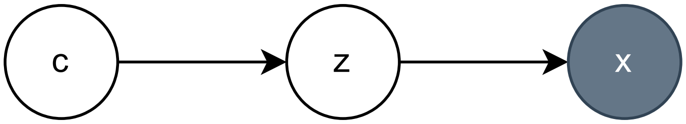


Figure 4.3: VaDE generative process. Note that after obtaining z for a given cluster, x is generated based on *only* z, independently of c.

density function of the GMM is defined as:

$$p(x) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k, \sigma_k^2 I) \tag{4.16}$$

where $\mathcal{N}(x|\mu_k, \sigma_k^2 I)$ is the probability density function of the Gaussian distribution with mean μ_k and variance σ_k^2. The notation $\sigma^2 I$ is a shorthand referring to a covariance matrix such that only marginal variances are non-zero, i.e. $\sigma^2 I := \text{diag}(\sigma_1^2, \sigma_2^2, ..., \sigma_o^2)$. The parameters of the GMM are typically learned using the **Expectation-Maximization** (**EM**) algorithm (Bishop (2006), Chapter 9.4), which iteratively estimates the parameters of the model by maximizing the expected likelihood of the data. In the context of VaDE, the GMM is first used to initialize the latent space distribution before training the core deep clustering model learning its own mixture-of-Gaussians distribution.

The core of the VaDE method is defined along the lines of the generative process used to generate the examples belonging to each of the clusters. The process to generate example x can be summarized as follows:

1. Choose a cluster $c \sim \text{Cat}(\boldsymbol{\pi})$ – categorical distribution with mixture coefficients $\boldsymbol{\pi} = [\pi_1, \pi_2, ..., \pi_K]$ for each of the K clusters,
2. Choose a latent representation $z \sim \mathcal{N}(\mu_c, \sigma_c^2 I)$ – Gaussian distribution with mean μ_c and covariance σ_c^2 corresponding to the chosen cluster c, similarly to the GMM framework,
3. Compute parameters μ_x and σ_x^2 for conditional distribution of x given z using a decoder f_θ (where θ is a parameter of the decoder network):

$$[\mu_x; \log \sigma_x^2] = f_\theta(z) \tag{4.17}$$

4. Choose an example $x \sim \mathcal{N}(\mu_x, \sigma_x^2 I)$.

The generative process described above is illustrated in Figure 4.3. As a consequence of the way the process is generated we remark that $x \perp\!\!\!\perp c|z$. Thus, the joint probability $p(x,z,c)$ is factorized as:

$$p(x,z,c) = p(c)p(z|c)p(x|z) \tag{4.18}$$

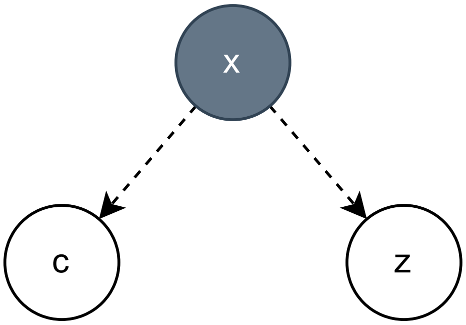


Figure 4.4: VaDE variational approximation. Note that both z and c are obtained from x independently.

where $p(c)$ is the prior distribution of the cluster, $p(z|c)$ is the prior distribution of the latent space given the cluster and $p(x|z)$ is the likelihood of the data given the latent representation. We can define those probabilities as:

$$\begin{aligned} p(c) &= \mathrm{Cat}(c|\boldsymbol{\pi}) \\ p(z|c) &= \mathcal{N}(z|\mu_c, \sigma_c^2 I) \\ p(x|z) &= \mathcal{N}(x|\mu_x, \sigma_x^2 I) \end{aligned} \tag{4.19}$$

VaDE model is trained by maximizing the log-likelihood of the observed data, which can be expressed as:

$$\log p(x) = \log \sum_{c=1}^{K} p(x,c) = \log \sum_{c=1}^{K} \int p(x,z,c)\,\mathrm{d}z \tag{4.20}$$

Assuming that $q(z,c|x)$ is the variational approximation to the true posterior $p(z,c|x)$, we can lower-bound the log-likelihood using the Jensen's inequality, similarly to the VAE case described in the previous Section, obtaining the evidence lower bound $\mathscr{L}_{\mathrm{ELBO}}(x)$:

$$\log p(x) \geq \mathbb{E}_{q(z,c|x)}\left[\log \frac{p(x,z,c)}{q(z,c|x)}\right] = \mathscr{L}_{\mathrm{ELBO}}(x) \tag{4.21}$$

In VaDE, $q(z,c|x)$ is additionally assumed to be a mean field distribution, factorized into $q(z,c|x) = q(z|x)q(c|x)$ – or equivalently, we have $z \perp\!\!\!\perp c|x$ (depicted in Fig. 4.4). Then ELBO can be then expressed as:

$$\begin{aligned}\mathscr{L}_{\text{ELBO}}(x) &= \mathbb{E}_{q(z,c|x)}\left[\log\frac{p(x,z,c)}{q(z,c|x)}\right]\\ &= \mathbb{E}_{q(z,c|x)}\left[\log\frac{p(x|z)p(z|c)p(c)}{q(z|x)q(c|x)}\right]\\ &= \mathbb{E}_{q(z,c|x)}\left[\log p(x|z)+\log p(z|c)+\log p(c)-\log q(z|x)-\log q(c|x)\right]\end{aligned} \tag{4.22}$$

In VaDE we model $q(z|x)$ using an encoder h_ϕ, where ϕ is a parameter of the encoder network:

$$\begin{aligned}[\widetilde{\mu};\log\widetilde{\sigma}^2] &= h_\phi(x)\\ q(z|x) &= \mathcal{N}(z|\widetilde{\mu},\widetilde{\sigma}^2 I)\end{aligned} \tag{4.23}$$

Encoding process is identical to VAE, and for optimization purposes, the reparametrization trick is used to example z from $q(z|x)$: $z=\widetilde{\mu}+\widetilde{\sigma}\cdot\epsilon$. Additionally, it can be shown (Jiang et al. 2017) that maximizing ELBO wrt $q(c|x)=q_\phi(c|x)$ yields that

$$q(c|x)=p(c|z)=\frac{p(z|c)p(c)}{\sum_{c'=1}^{K}p(z|c')p(c')} \tag{4.24}$$

The empirical equivalent of Equation (4.22) is (see Appendix A.1 for derivation):

$$\begin{aligned}\widehat{\mathscr{L}_{\text{ELBO}}}(x) = &-\frac{1}{2}\left(p\log 2\pi+\sum_{i=1}^{p}\left(\log\sigma_{x_i}^2+\frac{1}{L}\sum_{l=1}^{L}\frac{\left(x_i^{(l)}-\mu_{x_i}\right)^2}{\sigma_{x_i}^2}\right)\right)\\ &-\frac{1}{2}\sum_{c=1}^{K}\gamma_c\sum_{j=1}^{J}\left(\log\sigma_{c,j}^2+\frac{\widetilde{\sigma}_j^2}{\sigma_{c,j}^2}+\frac{(\widetilde{\mu}_j-\mu_{c,j})^2}{\sigma_{c,j}^2}\right)\\ &+\frac{1}{2}\sum_{j=1}^{J}\left(\log\widetilde{\sigma}_j^2+1\right)\\ &+\sum_{c=1}^{K}\gamma_c\log\frac{\pi_c}{\gamma_c},\end{aligned} \tag{4.25}$$

where L is the number of Monte Carlo examples for each feature vector used in reparametrization trick, J is the dimensionality of μ_c, σ_c^2, $\widetilde{\mu}_c$ and $\widetilde{\sigma}_c^2$ and γ_c denotes $q(c|x)$ for simplicity.

Note that ELBO from Equation (4.22) can also be rewritten alternatively as:

$$
\begin{aligned}
\mathscr{L}_{\text{ELBO}}(x) &= \mathbb{E}_{q(z,c|x)}\left[\log \frac{p(x,z,c)}{q(z,c|x)}\right] \\
&= \mathbb{E}_{q(z,c|x)}\left[\log \frac{p(x|z)p(z|c)p(c)}{q(z|x)q(c|x)}\right] \\
&= \mathbb{E}_{q(z,c|x)}[\log p(x|z)] + \mathbb{E}_{q(z,c|x)}\left[\log \frac{p(z|c)p(c)}{q(z|x)q(c|x)}\right] \\
&= \mathbb{E}_{q(z,c|x)}[\log p(x|z)] + \mathbb{E}_{q(z,c|x)}\left[\log \frac{p(z,c)}{q(z,c|x)}\right] \\
&= \underbrace{\mathbb{E}_{q(z,c|x)}[\log p(x|z)]}_{\text{reconstruction term}} - \underbrace{D_{\text{KL}}(q(z,c|x)||p(z|c))}_{\text{regularization term}}
\end{aligned}
\tag{4.26}
$$

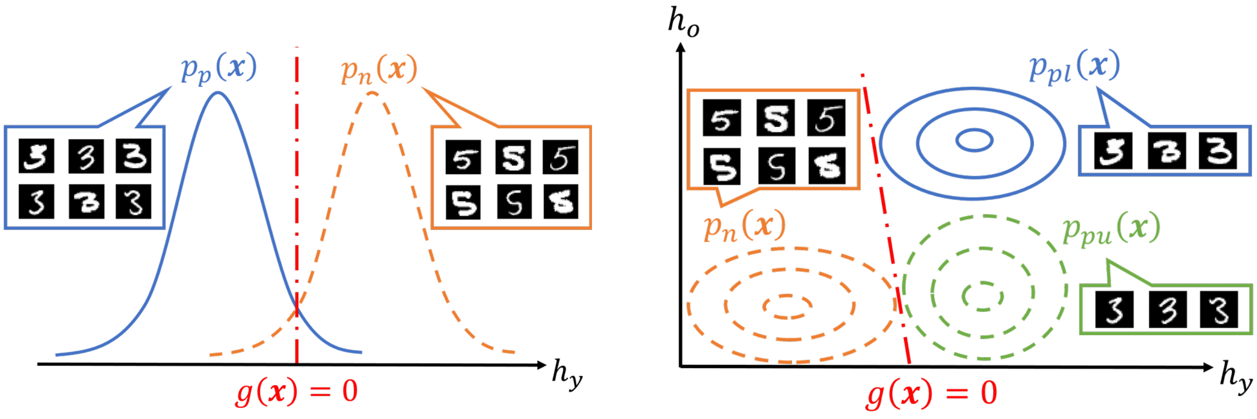
This form of ELBO is more similar to the one used in VAEs – the first term is the reconstruction loss, while the second term is the KL divergence between the variational posterior $q(z,c|x)$ and the prior $p(z|c)$ regularizing the latent space.

4.4 VAE-PU model

Strong clustering performance of VaDE combined with its cluster-specific example generation inspired development of many other methods based on similar architecture. VAE-PU method (Na et al. 2020) is one of such methods, designed to address the no-SCAR PU learning problem in the context of deep generative models. The core idea behind VAE-PU is to leverage the generative capabilities of VaDE to generate examples from the unknown positive-unlabeled data distribution, since in the no-SCAR scenario the distribution of labeled positives differs from that of positive examples present in the unlabeled dataset. Recall Equation (2.7) introduced as a part of Theorem 2.4, which represented the no-SCAR PU learning problem risk in the following form:

$$
\begin{aligned}
R(g) =& P(S=1)\,\mathbb{E}_{X|S=1}\, l(g(x)) \\
&+ P(Y=1,S=0)(\mathbb{E}_{X|Y=1,S=0}\,(l(g(x)) - l(-g(x))) \\
&+ P(S=0)\,\mathbb{E}_{X|S=0}\, l(-g(x))
\end{aligned}
\tag{2.7 revisited}
$$

Note that the above equality can be justified by the following reasoning: the first term in Equation (2.7) is a correct part of risk corresponding to an assignment of all labeled elements to the positive class, the third term corresponds to assigning all unlabeled items to the negative class, and the second term is the correction of the latter which accounts for the error committed in the case of positive unlabeled (PU) data which should be classified as positive. Note that $R(g)$ is not directly estimable as the second term involves calculation of the expected value with



(a) PU with SCAR

(b) PU with latent separation of observation and label

Figure 4.5: Motivation of separation on the latent variable of label(h_y)and the latent variabe of observation indicator (h_o, denoted as h_s in our paper) to overcome the Selected Completely At Random (SCAR). We illustrate the distribution that can be clarified from the data as the solid lines, and the unknown distribution from the data as the dotted lines. $g(x)$ is the label classifier that separates positive (p_p) and negative (p_n)distributions. If a bold font "3" is selectively labeled as positive, which violates SCAR, the data distributions of positive labeled (p_{pl}) and positive-unlabeled (PU) (p_{pu}) become different. The previous PU learning with SCAR does not distinguish between p_{pl} and p_{pu} as (a). Therefore, the model takes into account the observation indicators of labeling, as well as the labels. Figure and caption taken from Na et al. (2020).

respect to the distribution of the positive unlabeled cases, which is not observed. Its estimation will involve reconstruction of such elements. Namely, the empirical counterpart $R_{emp}(g)$ of $R(g)$ is

$$\begin{aligned} R_{emp}(g) = \frac{\pi_{PL}}{|\chi_{PL}|} \sum_{x^{(pl)} \in \chi_{PL}} l\left(g(x^{(pl)})\right) + \frac{\pi_{PU}}{|\widetilde{\chi}_{PU}|} \sum_{\widetilde{x}^{(pu)} \in \widetilde{\chi}_{PU}} l\left(g(\widetilde{x}^{(pu)})\right) \\ + \max\left\{0, -\frac{\pi_{PU}}{|\widetilde{\chi}_{PU}|} \sum_{\widetilde{x}^{(pu)} \in \widetilde{\chi}_{PU}} l\left(-g(\widetilde{x}^{(pu)})\right)\right. \\ \left. + \frac{\pi_U}{|\chi_U|} \sum_{x^{(u)} \in \chi_U} l\left(-g(x^{(u)})\right)\right\}, \end{aligned} \tag{4.27}$$

where χ_{PU}, χ_U and χ_{PL} denote positive unlabeled set, unlabeled set and positive labeled set, respectively, and $|A|$ stands for the set A size. Then, $\pi_{PL} = \widehat{P}(S=1)$, $\pi_{PU} = \widehat{P}(Y=1, S=0)$, $\pi_U = \widehat{P}(S=0)$, where $\widehat{P}$ denotes estimate of probability P. Importantly, $\widetilde{\chi}_{PU}$ in (4.27) denotes some estimate of positive unlabeled set χ_{PU} we are about to construct.

To obtain the reconstruction of the positive unlabeled data, VAE-PU uses the generative autoencoder inspired by VaDE approach. In order to achieve this, it is assumed in Na et al. (2020) that every observation has two characteristics, one responsible for its class assignment (positive or negative), and one connected with its labeling status (labeled or unlabeled). The intuition behind this assumption is depicted in Figure 4.5. This assumption motivates the use

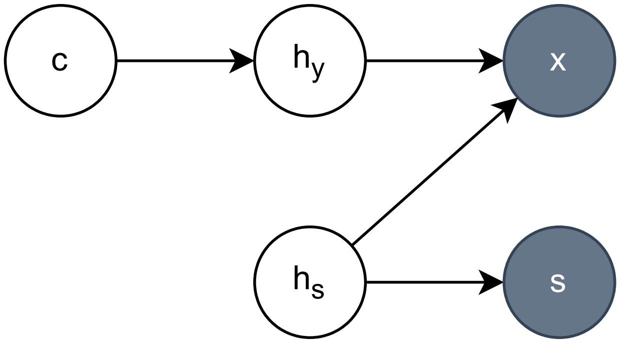


Figure 4.6: VAE-PU generative process. h_s does not depend on c, x is generated based on h_y and h_s, independently of c, and s is generated based on h_s.

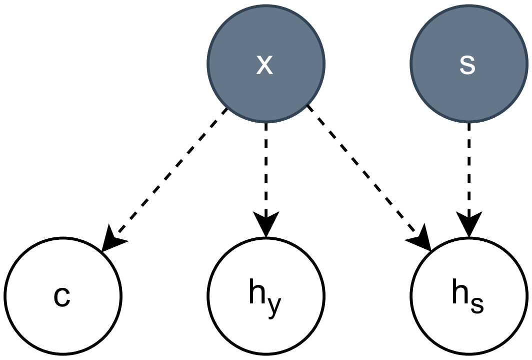


Figure 4.7: VAE-PU variational approximation. Note that h_y and c are obtained from x independently, and h_s is obtained from both x and s.

of a latent space consisting of two random components, where one (h_y) is responsible for the example's class assignment, and the other for its labeling status (h_s). The assumed probabilistic graphical model of the variational autoencoder is shown in Figure 4.6. VAE-PU preserves the VaDE clustering mechanism with number of clusters $K = 2$, and c becomes the indicator of the instance being positive regardless of its labeling, effectively distinguishing the positive and negative clusters. Variable h_y, as the latent class assignment representation, depends on c, while h_s is obtained independently of c. The example x is then generated based on h_y and h_s, as we assume that both of them characterize the observation, while its label s needs only information contained in h_s. The generative process of VAE-PU can be summarized as follows:

1. Choose a cluster $c \sim \text{Bern}(\pi)$ – now c is a binary variable indicating the positive/negative cluster,
2. Choose a latent representation $h_y \sim \mathcal{N}(\mu_c, \sigma_c^2 I)$ – latent class indicator representation for the chosen cluster c,
3. Choose a latent representation $h_s \sim \mathcal{N}(0, I)$ – independent of the cluster assignment,
4. Compute μ_x and σ_x^2 using a decoder f_θ (where θ is a parameter of the decoder network):
$$[\mu_x; \log \sigma_x^2] = f_\theta(h_y, h_s)$$
5. Choose an example $x \sim \mathcal{N}(\mu_x, \sigma_x^2 I)$,
6. Decode labeling status distribution parameter using the observation classifier f_{s_ξ} (where ξ is a parameter of the observation classifier network):
$$r_s = f_{s_\xi}(h_s)$$
7. Assign a label $s \sim \text{Bern}(r_s)$.

The generative process above leads to the following joint probability factorization:

$$p(x, h_y, h_s, c, s) = p(c)p(h_y|c)p(h_s)p(x|h_y, h_s)p(s|h_s) \tag{4.28}$$

where $p(c)$ is the prior distribution of the cluster, $p(h_y|c)$ is the prior distribution of the latent class assignment given the cluster, $p(h_s)$ is the prior distribution of the labeling status, $p(x|h_y, h_s)$ is the likelihood of the data given both of latent representations, and $p(s|h_s)$ is the likelihood of data labels given the labeling status representation. In VAE-PU case these distributions can be defined as:

$$\begin{aligned}
p(c) &= Bern(c|\pi) \\
p(h_y|c) &= \mathcal{N}(h_y|\mu_c, \sigma_c^2 I) \\
p(h_s) &= \mathcal{N}(h_s|0, I) \\
p(x|h_y, h_s) &= \mathcal{N}(x|\mu_x, \sigma_x^2 I) \\
p(s|h_s) &= Bern(s|r_s)
\end{aligned} \tag{4.29}$$

Similarly to VaDE, the variational approximation $q(h_y, h_s, c|x, s)$ to the true posterior $p(h_y, h_s, c|x, s)$ is introduced in order to follow the variational inference paradigm. It is assumed to be a mean field distribution, i.e. $c \perp\!\!\!\perp h_y \perp\!\!\!\perp h_s | x, s$ (see Fig. 4.7), and we also assume that the class related variables (h_y and c) are independent of the label s, which leads to the following factorization:

$$\begin{aligned} q(h_y, h_s, c|x, s) &= q(h_y|x, s)q(h_s|x, s)q(c|x, s) \\ &= q(h_y|x)q(h_s|x, s)q(c|x) \end{aligned} \tag{4.30}$$

After a derivation similar to VaDE, we obtain ELBO for VAE-PU model (cf (14) in Na et al. (2020)):

$$\begin{aligned} &\mathscr{L}_{\text{ELBO}}(x, s) \\ &= \mathbb{E}_{q(h_y, h_s, c|x,s)} \log \frac{p(h_y, h_s, c, x, s)}{q(h_y, h_s, c|x, s)} \\ &= \mathbb{E}_{q(h_y, h_s, c|x,s)} \big[\log p(x|h_y, h_s) + \log p(h_y|c) + \log p(s|h_s) + \log p(h_s) + \log p(c) \\ &\quad - \log q(h_y|x) - \log q(h_s|x, s) - \log q(c|x) \big] \end{aligned} \tag{4.31}$$

and its empirical equivalent (see Appendix A.2 for derivation):

$$\begin{aligned} &\widehat{\mathscr{L}}_{\text{ELBO}}(x, s) \\ &= -\frac{1}{2}\left(p \log 2\pi + \sum_{i=1}^{p} \left(\log \sigma_{x_i}^2 + \frac{1}{L} \sum_{l=1}^{L} \frac{\left(x_i^{(l)} - \mu_{x_i}\right)^2}{\sigma_{x_i}^2} \right) \right) \\ &\quad - \frac{1}{2} \sum_{c \in \{0,1\}} \gamma_c \sum_{j=1}^{J_y} \left(\log \sigma_{c,j}^2 + \frac{\sigma_{y,j}^2}{\sigma_{c,j}^2} + \frac{(\mu_{y,j} - \mu_{c,j})^2}{\sigma_{c,j}^2} \right) \\ &\quad + \frac{1}{2} \sum_{j=1}^{J_y} \left(1 + \log \sigma_{y,j}^2 \right) \\ &\quad - \frac{1}{2} \sum_{j=1}^{J_s} \left(1 + \log \sigma_{s,j}^2 - \sigma_{s,j}^2 - \mu_{s,j}^2 \right) \\ &\quad + \sum_{c \in \{0,1\}} \gamma_c \log \frac{\pi_c}{\gamma_c} \\ &\quad + s \log \widetilde{s} + (1 - s) \log(1 - \widetilde{s}), \end{aligned} \tag{4.32}$$

where L is the number of Monte Carlo examples for each feature vector used in reparametrization trick, J_y and J_s are the dimensionalities of latents h_y and h_s respectively, $\widetilde{s}$ is the output of the observation classifier, μ_* and σ_* denote output mean and variance of the encoder of variable $*$, and γ_c denotes $q(c|x)$ for simplicity. Note that Equation (4.32) has a different form than the one presented in Na et al. (2020) – reconstruction term used by Na et al. is adapted to the binary situation, where the modelled distribution is a product of D independent Bernoulli distributions, which, while simpler in form, is rarely used in practice.

To ensure that the generated positive unlabeled instances will be representative of the distribution of positive unlabeled set, during the VAE-PU training two additional losses are used in Na et al. (2020) in addition to the ELBO. The first one is the **adversarial generation loss** $\mathscr{L}_{\text{adv}}$ utilizing a discrimination network D separate from the rest of VAE-PU, defined as:

$$\mathscr{L}_{\text{adv}} = \mathbb{E}_{x \sim p_U(x)} \log D(x) + \mathbb{E}_{x \sim p_{PU}(x)} \log(1 - D(x)) \tag{4.33}$$

and its empirical counterpart:

$$\begin{aligned}\widehat{\mathscr{L}_{\text{adv}}} =& \frac{1}{|\chi_U|} \sum_{x^{(u)} \in \chi_U} \log D(x^{(u)}) \\ &+ \frac{1}{|\widetilde{\chi}_{PU}|} \sum_{\widetilde{x}^{(pu)} \in \widetilde{\chi}_{PU}} \log\left(1 - D\left(\widetilde{x}^{(pu)}\right)\right).\end{aligned} \tag{4.34}$$

By optimizing this loss, we ensure that the generated positive unlabeled instances are close to the unlabeled instances. The second additional loss is the **target loss** (called label loss in Na et al. (2020)) $\mathscr{L}_{\text{target}}$, utilizing the target classifier g:

$$\mathscr{L}_{\text{target}} = \mathbb{E}_{x \sim p_{PU}(x)}\, l(g(x)) \tag{4.35}$$

with its empirical counterpart being:

$$\widehat{\mathscr{L}_{\text{target}}} = \frac{1}{|\widetilde{\chi}_{PU}|} \sum_{\widetilde{x}^{(pu)} \in \widetilde{\chi}_{PU}} \left(g\left(\widetilde{x}^{(pu)}\right)\right). \tag{4.36}$$

The target loss is used to ensure that the generated positive unlabeled instances are close to the true positive instances.

The final objective function of VAE-PU is then defined as:

$$\min_G \max_D (-\mathscr{L}_{\text{ELBO}} + \alpha \mathscr{L}_{\text{adv}} + \beta \mathscr{L}_{\text{target}}) \tag{4.37}$$

where α and β are hyperparameters controlling the importance of the adversarial generation loss and the target loss, respectively, G consists of the variational encoder, decoder and observation classifier networks of the VAE-PU, and D is the discriminator network. We kept the values of the hyperparameters consistent with the ones used by Na et al. (2020). Due to the adversarial generation loss, the optimization becomes a min-max problem. $\mathscr{L}_{\text{ELBO}}$ specifies how much VAE-PU can represent the input data, and $\mathscr{L}_{\text{adv}}$ brings the generated PU instances closer to unlabeled, and $\mathscr{L}_{\text{target}}$ – positive, distributions respectively.

Given the trained VAE-PU model, we can generate an artificial positive unlabeled dataset $\widetilde{\chi}_{PU}$ in the following way:

1. Match positive and chosen unlabeled examples into pairs. The are multiple ways to approach this problem, and the one used throughout this thesis is matching the examples with nearest h_y representation – see Section 4.5.2 for further discussion of example matching,
2. For each pair, extract latent label information from positive instance $\left(h_y^{(pl)}\right)$ and latent observation status from unlabeled set $\left(h_s^{(u)}\right)$,
3. Concatenate $h_y^{(pl)}$ and $h_s^{(u)}$ to obtain the combined latent representation,
4. Decode the latent representation using decoder f_θ to obtain the generated positive unlabeled instance.

The examples generated this way are then used in the second step of the training, optimizing the target classifier g:

$$\min_g R_{emp}(g), \tag{4.38}$$

where $R_{emp}(g)$ is defined in (4.27) That allows us to obtain the final classifier g which can be used for future predictions.

4.5 Modifications of VAE-PU

Although generative approach of VAE-PU proposed in Na et al. (2020) has significant advantages, there are several aspects of it amenable to improvements, both in the model formulation and implementation details. In the following we discuss some modifications which we introduced to the original version of VAE-PU[1]. We recall that the prior probability π is assumed to be known. The modifications described here provide the basis for all of the VAE-PU based methods presented in this thesis: VAE-PU+OCC, the first model we introduce later in this chapter, as well as VAE-PU+FOR and VAE-PU-Bayes, which will be presented in Chapters 5 and 6, respectively.

4.5.1 Loss function

The original function used by VAE-PU is the inverse sigmoid loss: $l_{\text{sig}}(z) = (1+e^z)^{-1}$. Although this function is monotone, theoretical risk $\mathbb{E}\ l_{\text{sig}}(Yg(X))$ corresponding to it has no stationary point when $P(Y=1|x) \not\equiv \frac{1}{2}$ (i.e. g^* such that $\nabla R(g^*) = 0$ does not exist, see Appendix B.3 for more details) whereas in the case of logistic loss $l_{\text{logistic}}(z) = \log(1+e^{-z})$ the function being

[1] Available at GitHub: `https://github.com/wp03052/vae-pu`

Table 4.1: Comparison of different item matching algorithms, MNIST 3v5 dataset

Dataset	Mean digit boldness
PL	0.2475
U	0.1346
True PU	**0.1397**
Generated PU – $\boldsymbol{h_s}$-matching	**0.2255**
Generated PU – $\boldsymbol{h_y}$-matching	**0.1451**

the stationary point g^* of the risk is a monotone function of odds of posterior probability $g^*(X) = P(Y = 1|X)/P(Y = -1|X)$. Thus the corresponding classification rule is a Bayes rule (see Appendix B.1). Consequently, l_{sig} has heuristic justification only, whereas l_{logistic} has a strong theoretical underpinning. This is a main reason we have replaced the inverse sigmoid loss by the logistic loss in our implementation.

4.5.2 PU example generation process

During example generation process, positive observations have to be matched with chosen unlabeled examples. Using values of latent variable h_y for labeled data, denoted by $h_y^{(pl)}$ we can construct pseudo-dataset pertaining to P_{PU} in the following two ways:

(i) for each element of labeled dataset with corresponding vector $\left(h_y^{(pl)}, h_s^{(pl)}\right)$, one finds an element among unlabeled examples with $\left(h_y^{(u)}, h_s^{(u)}\right)$ such that $h_y^{(u)}$ is the closest to $h_y^{(pl)}$ wrt the euclidean distance (h_y-matching) and creates a latent vector $\left(h_y^{(pl)}, h_s^{(u)}\right)$ which mimics latent $\left(h_y, h_s\right)$ vector of a potential positive unlabeled observation. From this vector a pseudo-observation from positive unlabeled (PU) population is reconstructed by decoder part of VAE,

(ii) the above procedure is repeated, but now the matching element from unlabeled data is chosen by picking $\left(h_y^{(u)}, h_s^{(u)}\right)$ such that $h_s^{(u)}$ is closest to $h_s^{(pl)}$ (h_s-matching). The remaining part of the construction is the same.

Similarly to the choice of the specific loss function, VAE-PU does allow for arbitrary way of estimating the PU class. In particular, it can be based on matching process described above. In the paper (Na et al. 2020), it is suggested that the items are matched according to (i) method described above i.e. based on the distances in h_y latent space – each item in P set is matched with the item from the U set with the closest h_y representation.

However, in the actual implementation, distances in the h_s space were used instead, as described in (ii) above. We believe that generating PU examples in that way leads to deterioration of the generative properties of the VAE-PU. Results of one of the tests conducted by us, illustrating the generation process on the MNIST 3v5 dataset, are shown in Table 4.1. Items were labeled based on the digit boldness, with bolder digits having a higher chance on being labeled. It is apparent that h_y-based example matching leads to generation of the examples more similar (in terms of boldness) to the true PU set than h_s-based example matching, where the generated items were more similar to PL set. It is the reason why in the modified algorithm we used h_y-based item matching.

4.5.3 Max term introduction in empirical VAE-PU risk

The issue we explain now is our main motivation for VAE-PU to be modified using one-class classification approach. Consider the reason why max term has been introduced in Equation (4.27)[2]. This is mainly due to the fact that the observations in χ_{PU} are replaced by pseudo-observations in $\widetilde{\chi}_{PU}$, *which do not form a subset of* χ_U and thus the estimator of of a *nonnegative* summand of the risk equal to

$$\mathbb{E}\, l(g(-X))\mathbb{I}\{S=0\}\, \mathbb{E}\, l(g(-X))\mathbb{I}\{S=0, Y=1\} \tag{4.39}$$

is not necessarily positive. For case-control PU truncation at 0 results in significant improvement for the corrected estimator nnPU over its un-corrected version uPU (Sugiyama et al. (2022), Chapter 11). Truncation leads to a substantial loss of information, however. Truncation is meant to decrease bias of $R_{emp}(g)$ and, of course, it does the trick when we know that the theoretical counterpart of the term we replace by 0 is necessarily non-negative. However, once the term is truncated by 0, we can not modify it to make it asymptotically unbiased for the respective theoretical term. We argue that the truncation is not necessary if the set of pseudo-observations $\widetilde{\chi}_{PU}$ is replaced by the subset of observations from χ_U which is similar to χ_{PU} . Indeed, suppose for a moment that $\widetilde{\chi}_{PU}$ in (4.27) is replaced back by the true χ_{PU}. The corresponding part of the empirical risk is

$$-\frac{\pi_{PU}}{|\chi_{PU}|}\sum_{x^{(pu)}\in\chi_{PU}} l\left(-g(x^{(pu)})\right)+\frac{\pi_U}{|\chi_U|}\sum_{x^{(u)}\in\chi_U} l\left(-g(x^{(u)})\right). \tag{4.40}$$

The expression above is bound to be positive as in view of Law of Large Numbers $\pi_{PU}/|\chi_{PU}| \approx \pi_U/|\chi_U| \approx n^{-1}$ and the second sum in the above expression is larger then the first sum *for original data*. Thus, were $\widetilde{\chi}_{PU}$ a subset of χ_U, satisfying the approximate weight equality, introduction of max correction would not be necessary. Thus, our aim is to determine a subset of χ_U which would correspond to positive unlabeled observations. We show that instead of using the generated PU items directly, it is substantially more beneficial to take advantage of the generated dataset to find

[2] Our modified implementation corrected the error in the original code (related to max term application) which deviated from the original formula (4.27).

the observations which are likely to be true-PU items *in unlabeled dataset*. The remaining part of this chapter will be dedicated to the discussion of the ways to achieve this goal by one-class classification and the results of such an approach.

In order to apply one-class classification we will treat $\widetilde{\chi}_{PU}$, constructed as described at the end of Section 4.4 according to the h_y matching described in Section 4.5.2, as the sample from normal population described by P_{PU} distribution and χ_U as the sample corresponding to the mixture of P_{PU} and P_N.

4.6 VAE-PU+OCC – One-class classification enhancement of VAE-PU

4.6.1 One-Class Classification OCC

One-class classifiers (OCC) are a family of methods which, given a training dataset drawn from some normal distribution P_X, test which of the new items are outliers or anomalies, in the sense that they are drawn from a different distribution than the normal one; for a recent review see Ruff et al. (2021). This task is also frequently known as anomaly (or novelty, outlier, out of distribution) detection, or learning from the positive class only (Moya, Koch, and Hostetler 1993; Pimentel et al. 2014). There are many practical situations where such scenario occurs e.g. medical analysis, fraud detection and forensic analysis. Note that there is a substantial difference between one-class classification and biased PU problem. In contrast to one-class classification when unbiased sample from positive class is available, in the latter case, as has been said, we observe only biased observations from the positive class and the general population. Nevertheless, we are able to reduce biased PU problem to one-class classification problem, by treating $\widetilde{\chi}_{PU}$ as the sample from the normal population pertaining to P_{PU} and χ_U as the sample generated from the mixture of P_{PU} and P_N. Note that apart from the fact that $\widetilde{\chi}_{PU}$ are generated from the distribution which is only close to P_{PU}, the second difference between PU and one-class problems consists in that our primary objective is to detect normal data, not anomalies in χ_U.

Usually, the one-class classification methods output score value for each new example. We mention in this context GAN-based methods which use the reconstruction loss as an anomaly score, compare e.g. ADGAN algorithm (Schlegl et al. 2017). This is in contrast to classification, where often we can interpret the results in terms of posterior probability or class assignment. This score based approach makes evaluation of new data difficult to handle – for instance, defining a decision function (in practice usually via a threshold) is difficult; for some methods (e.g. One-Class SVM) such a boundary might be defined naturally, but many others fail to give any statistical guarantees on their outputs. Several approaches exist to tackle this issue (Vovk, Gammerman, and Saunders 1999; Vovk, Gammerman, and Shafer 2005); here we use p-values for scores based on validation set which ensure that p-value for observation stemming from normal population will be super-uniform and thus probability of false signal (i.e. erroneous detection of an outlier) can be easily controlled (Bates et al. 2023).

We will now discuss some specific one-class methods we used in our VAE-PU+OCC algorithm. We stress that our aim is not to construct a new OCC method but to verify how the representative examples of the existing ones perform in our task. The classical one-class classification methods include One-Class SVM (OC-SVM) and Isolation Forest. In One-Class SVM (Schölkopf et al. 2001) approach, the coordinate center is treated as the only anomalous observation and hyperplane is sought with maximum margin separation from it for data from the normal class. Isolation Forest (Liu, Ting, and Zhou 2008) is based on an idea that anomalies can be detected in random forests (thus one uses random subsets of the data and random sets of features) by finding the leaves corresponding to the shortest paths in trees constituting the forest. Recently, ECOD (Empirical Cumulative distribution based Outlier Detection) and A^3 method has been proposed. ECOD (Li, Zhao, et al. 2022) is based on a premise that anomalies of multivariate distributions usually exhibit atypical behavior for marginal distributions of this distribution corresponding to one or several dimensions and thus the constructed anomaly measure has similar motivation to Fisher test statistic (see e.g. Bates et al. (2023)). Activation Anomaly Analysis $\left(A^3\right)$ method (Sperl, Schulze, and Böttinger 2021) is a significantly more complex approach based on neural networks. It employs three components: *target network*, which performs a task unrelated to anomaly detection (for example, autoencoder); *anomaly network*, generating anomalous examples based on input (in the simplest form, it might be a random sample generator, similarly to GAN); and *alarm network*, which discerns normal and anomalous examples based on hidden activations of the target network. This method is designed to work in unsupervised setting, but including some anomalous examples was shown to improve results.

4.6.2 VAE-PU+OCC – method introduction

The main idea of VAE-PU+OCC method is a straightforward one. The approach treats pseudo-set $\widetilde{\chi}_{PU}$ as generated from the normal class, and uses the fact that χ_U is a set consisting of a mixture of positive unlabeled elements (which are similar to elements in $\widetilde{\chi}_{PU}$) and negative elements which are considered as anomalies. One-class classifiers are now used to screen anomalies (elements with $Y = 0$) from regular elements (elements such that $Y = 1$ and $S = 0$).

The main idea of VAE-PU+OCC is combining the VAE-PU powerful generative capabilities and one-class classification. That means:

- The task is to find Positive Unlabeled (abbreviated as PU) observations χ_{PU} in the unlabeled dataset,

- Due to the biased labeling, Positive Labeled (PL) set χ_{PL} may not be used for this task as a benchmark representative, as it has a different distribution from that of PU observations χ_{PU} contained in the Unlabeled dataset χ_U,

- Instead of using unavailable Positive (P) set χ_P, we will train the classifier on PU pseudo-observations generated by VAE-PU itself,

- As we know PU pseudo-observations only, and we do not have any information on distribution of Negative (N) observations χ_N, we apply one-class classifier instead of the traditional binary classification method. This classifier is be used to filter the true-PU items out of the unlabeled dataset (which is a mixture of PU and N instances).

VAE-PU+OCC method applies a learned VAE-PU model. That means training all elements of the model – both the generative part (encoder, decoder, observation classifier, discriminator) and the target classifier itself. The trained target classifier is needed to include label loss in process of VAE-PU training. Using target classifier pretrained in that way instead of reinitializing the model is also beneficial for the quality of final model.

The final part of the training starts with generation of the PU pseudo-observations. Then, an OCC classifier of choice is trained on the generated data – as noted before, the method does not require any specific model. In this step, generated data is split into two (in our case, equal) parts: training and calibration; training set is used to train OCC model, while calibration dataset is reserved for subsequent p-value calculation. Then, based on the trained OCC model, all of the observations in the unlabeled set are evaluated. Resulting scores are then converted to marginal p-values by calculating the fraction of scores from the (pseudo)-normal observations exceeding the score of an item examined. Based on the class prior provided to VAE-PU, we can then evaluate the proportion of the PU items in the U dataset. Instead of using a predefined cutoff, we can then take observations from unlabeled dataset corresponding to the largest p-values and use that as an input to the VAE-PU risk function. This procedure is summed up in Algorithm 2. We stress again that using true-PU examples also solves the issue of loss reduction exceeding the original loss, which means that the max term in risk function $R_{emp}(g)$ is no longer necessary.

Algorithm 2 contains only a general outline of the training procedure. Two of the more specific improvements we used in our implementation are as follows:

1. In order to improve the diversity of the training set for the OCC classifier, the generation process is repeated several times. Due to inherent randomness of the decoder, generated observations will be slightly different in each generated batch. In our case, the process was repeated until generated set size was equal to original dataset size,

2. We implemented early stopping to avoid overfitting the OCC procedure. Using OCC to train VAE-PU target classifier usually causes its precision to increase, but the recall often

Algorithm 2: VAE-PU+OCC training

```
Input: $\pi$ – class prior, $n$ – number of training items
Train VAE-PU model (encoder, decoder, target classifier, observation classifier and
  discriminator); this process is described in detail in Section 4.4;
while not converged do
    Generate pseudo-set $\widetilde{x}_{PU}$ using trained VAE;
    Train OCC classifier of choice on $\widetilde{x}_{PU}$;
    Use OCC classifier to calculate marginal p-values for U dataset;
    Calculate estimated proportion $p$ of PU examples in U corresponding to
      $P(Y = 1|S = 0)$:
        $p = \frac{\pi - \frac{n_{S=1}}{n}}{\frac{n_{S=0}}{n}} = \frac{n\pi - n_{S=1}}{n_{S=0}}$;
    Choose proportion $p$ of all examples in U with the largest p-values as the candidate
      PU set;
    Update VAE-PU target classifier with the risk function $R_{emp}(g)$ and candidate PU set;
end
```

decreases slightly as a tradeoff. In order to balance both the precision and the recall values, early stopping metric was the F1-score on validation dataset. Procedure iteration limit was set to 100 iterations, and early stopping usually decreased this to 10 to 20 epochs. Some cases required only a few (3-5) iterations, but the core Algorithm 2 hardly ever repeats more than 50 times.

It is also important to emphasize that there are several limitations on performance of the OCC variant:

- Dependence on generated set quality (or, in general, baseline VAE-PU performance) – when more adequate observations are generated, OCC is trained on more representative dataset and its predictions become more accurate,

- Dependence on label frequency which is due to the VAE-PU nature of generation process (i.e. matching each positive example to its closest neighbor in the latent space). When there are few labeled examples, regardless of the process repeats, the amount of information in the generated PU pseudo-set is also limited – all of the generated examples are based on this small set of PL items.

4.7 Experiments

4.7.1 Experimental setup

We assessed VAE-PU+OCC performance on several datasets to prove effectiveness of the OCC-based example selection for construction of classifiers. Four benchmark datasets resulting in 6 different tasks were used:

- **MNIST**[3] – two different tasks, 3 versus 5 (images of digit *3* are considered positive, *5* – negative, abbreviated to 3v5) and OvE (images of *odd* digits are positive, *even* – negative),
- **CIFAR-10**[4] – two different tasks, Car versus Truck (*automobile* images are positive, *truck* – negative) and Machine versus Animal (*airplane*, *automobile*, *ship* and *truck* images are positive, *bird*, *cat*, *deer*, *dog*, *frog* and *horse* – negative),
- **STL-10**[5] – identical classes (but more complex images) as in CIFAR-10, Machine versus Animal split is only considered,
- **Gas Concentrations**[6] – *Ethanol* examples are positive, *Ammonia* – negative.

Detailed description of the datasets is given in the Appendix E – there we also described labeling schemes applied in order to construct artificially labeled datasets from those above, ensuring correct no-SCAR problem modeling.

Main objective of this chapter is measuring the improvement provided by OCC-based example selection on VAE-PU learning. To this end, performance of VAE-PU+OCC was compared to baseline VAE-PU. The VAE-PU model was reimplemented in order to incorporate major performance improvements, which allowed for study of increased array of experiments. Two implementations were prepared (as included in the result tables), one incorporating modifications described at the end of Section 4.6.2, and the *orig* version, which preserves model settings from the original paper. As VAE-PU was shown to outperform multiple models (uPU, nnPU, PUbN\N, GenPU, PAN, PUSB; (Na et al. 2020), note that some of those models are designed for case control scenario) – those comparisons will not be repeated here, but instead two additional methods will be considered: SAR-EM (Bekker and Davis 2018b) and LBE (Gong et al. 2021). These two methods reflect a current state-of-the-art in biased PU classification. We also note that our proposed method and SAR-EM method are similar in that both are Empirical Risk Minimization methods and thus it is especially worthwhile to compare their performance with the proposed

[3] `http://yann.lecun.com/exdb/mnist/`

[4] `https://www.cs.toronto.edu/~kriz/cifar.html`

[5] `https://cs.stanford.edu/~acoates/stl10/`

[6] `https://archive.ics.uci.edu/ml/datasets/gas+sensor+array+drift+dataset`

method. In case of SAR-EM, we use the implementation provided by the authors[7]. In order for the algorithm to work, it needs a list of data attributes on which propensity score potentially depends – in our case, all attributes will be considered as such. We also prepared a custom implementation of LBE-LF architecture (Gong et al. 2021) ("LF" stands for Logistic Function version of the model).

VAE-PU+OCC model allows for an arbitrary choice of embedded one-class classifier. We tested four OCC models: One-Class SVM (Schölkopf et al. 2001), Isolation Forest (Liu, Ting, and Zhou 2008), A^3 (Sperl, Schulze, and Böttinger 2021) and ECOD (Li, Zhao, et al. 2022). It is important to note that in the experiments a slightly modified version of ECOD was used. In the official implementation[8] training and test dataset are concatenated, and then used to calculate ECDF during prediction. The modified version uses only training data to calculate ECDF for future predictions. Original VAE-PU paper (Na et al. 2020) considered only very low label frequencies (e.g. $c = 0.02$ for MNIST datasets). Although it is important to consider scenarios where training data information is severely limited, such a task is very difficult, especially considering its no-SCAR nature, and it is also rare that the training datasets exhibit that large L-U set imbalance. That lead us to expanding the test cases to the multitude of label frequency values, including the larger ones; for each dataset, five different label frequency values are considered: $c \in \{0.02, 0.1, 0.3, 0.5, 0.7\}$. For each label frequency, dataset and method the training and evaluation procedure is repeated 10 times, each time with different random seed equal to experiment number. Each such experiment is performed with a training-validation-testing split depending on the run number (preserving a constrant 70-15-15 ratio). We evaluated classification performance in terms of widely used metrics:

$$\begin{gathered}
Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \\
Precision = \frac{TP}{TP + FP}, \\
Recall = \frac{TP}{TP + FN}, \\
F1\ score = \frac{2 * Precision * Recall}{Precision + Recall} = \frac{2 * TP}{2 * TP + FP + FN}.
\end{gathered} \tag{4.41}$$

Code for modified VAE-PU (called baseline in the following) as well as its *orig* version, VAE-PU+OCC and performed experiments is publicly available at GitHub[9]. The repository also contains detailed instruction for computational experiment reproduction, including all software packages and their versions.

[7] https://github.com/ML-KULeuven/SAR-PU

[8] Implemented in PyOD: https://github.com/yzhao062/pyod/blob/master/pyod/models/ecod.py

[9] https://github.com/wawrzenczyka/VAE-PU-OCC

4.7.2 Motivational example

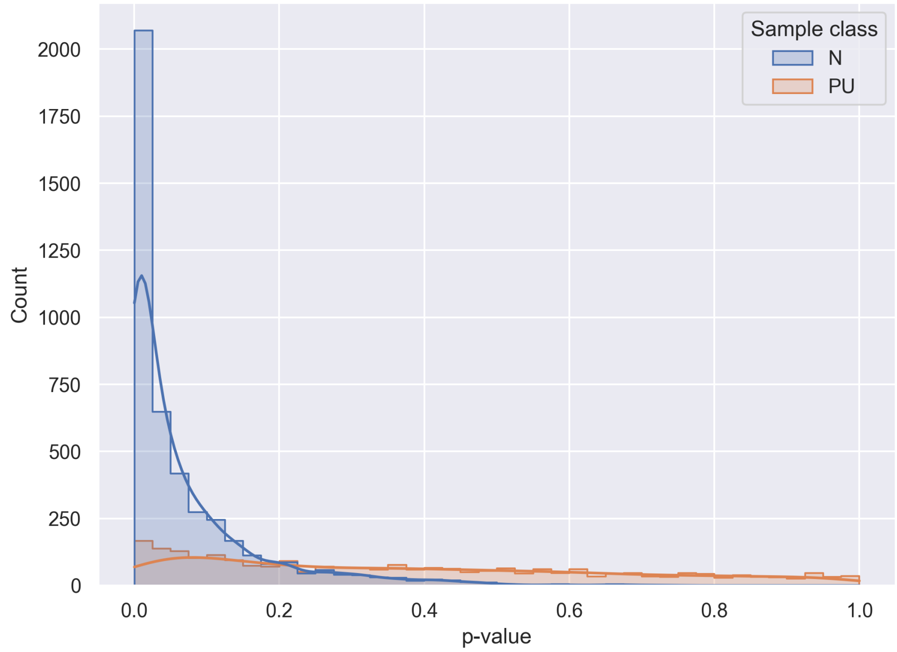
Before examining efficiency of the combined VAE-PU+OCC algorithm, we examine its one specific aspect. Namely, we will illustrate one-class classifier potential in discerning positive examples in unlabeled data. Consider initial steps of Algorithm 2 as follows: using generated PU set $\widetilde{\chi}_{PU}$ obtained via trained VAE-PU model, we train OCC classifier; then, such a classifier is evaluated on each observation in unlabeled dataset. Figure 4.8 depicts distribution of p-values obtained by A^3 classifier in this scenario on two datasets. MNIST 3v5 (Fig. 4.8a) is a straightforward example, where both distributions behave as expected; negative examples tend to have very low p-values, whereas PU p-value distribution is approximately uniform. CIFAR MachineAnimal (Fig. 4.8b) is an example of dataset which proved hard for most of the tested OCC methods; we can see that even though most of the true negative items are concentrated around 0, there are multiple cases when their p-value is really high, whereas positive examples are even more skewed, with almost all of their p-values being close to 1. Even though the latter dataset shows that the separation of PU and N parts of unlabeled dataset can be often imperfect, *overall* ability of OCC methods to find positive examples in U set is still remarkable. This serves as a basis of the following experiments, which focus on overall classification performance of VAE-PU+OCC.

4.8 Results

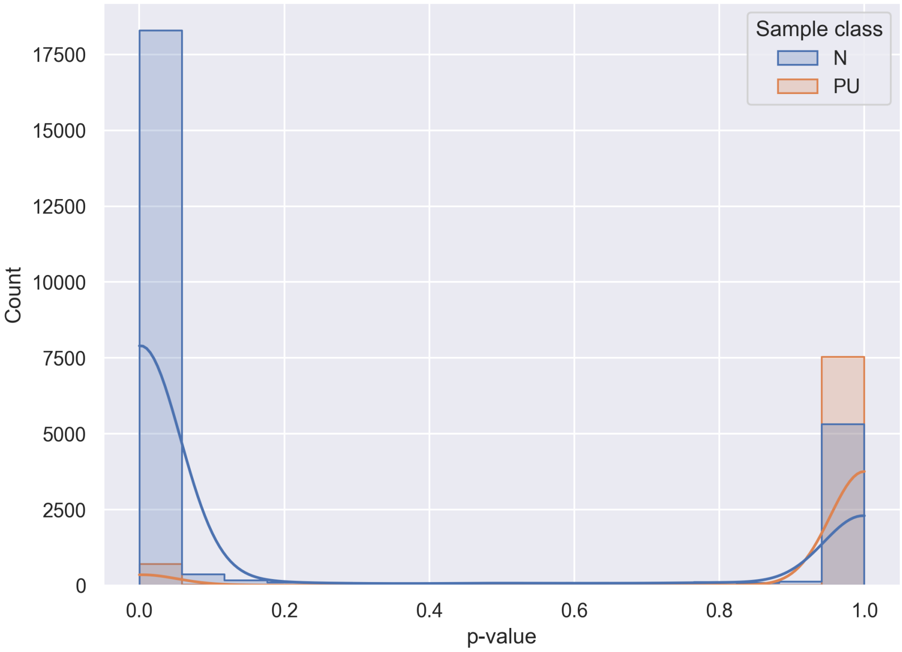
4.8.1 No-SCAR results

Tables 4.2 and 4.3 summarize experiments described in Section 4.7.1. For each experimental setting, given as a combination of dataset, label frequency and a particular method, we report the mean accuracy and F1 score, as well as the standard error of the respective metric. In each table results of the benchmark methods (two VAE-PU versions, denoted as "Baseline (original)" and "Baseline (modified)", SAR-EM, and LBE) are separated from the proposed OCC variants based either on A^3 method, ECOD, Isolation Forest or One-Class SVM.

It is apparent that SAR-EM and LBE methods are significantly outperformed, both by base VAE-PU and its modifications. This is especially pronounced in low label frequency setting, but remains true even when label frequency increases. This strongly suggests that it is beneficial to construct observation which mimic those from PU class, especially when size of positive set is small. Notable exceptions are high label frequency experiments (for $c = 0.7$) on CIFAR MachineAnimal, where SAR-EM outperformed (but barely) VAE-PU+OCC variants in terms of accuracy, and STL, where even though it achieved the highest accuracy for $c = 0.7$, its F1 score is significantly smaller than several OCC-based models. LBE method, on the other hand, managed to reach significantly better accuracy and F1 score on Gas Concentrations dataset (for



(a) MNIST 3v5



(b) CIFAR MachineAnimal

Figure 4.8: Distribution of p-values obtained by evaluation of trained A^3 classifier on unlabeled (U) dataset, separated by true example class.

Table 4.2: Accuracy values per dataset. Green ticks („✓") correspond to the cases when t-test rejected equality of accuracies of the VAE-PU-OCC method considered and that of the baseline original method in favor of the former one at $\alpha = 0.05$. Dashes („–") indicate the failure to reject (see Section 4.8.3). Bold entries correspond to maximum mean accuracy for a given dataset and label frequency combination.

c	Method	MNIST	MNIST	CIFAR	CIFAR	STL	Gas
		3v5	OvE	CarTruck	MachineAnimal	MachineAnimal	Concentrations
0.02	Baseline	79.99 ± 1.04	71.55 ± 1.03	78.63 ± 2.91	87.71 ± 1.03	75.82 ± 0.52	78.89 ± 2.78
	Baseline (orig)	78.18 ± 0.97	60.62 ± 0.60	79.36 ± 2.51	87.45 ± 1.59	73.62 ± 0.37	71.38 ± 3.16
	LBE	47.78 ± 0.29	49.65 ± 0.15	50.06 ± 0.39	60.20 ± 0.22	60.08 ± 0.27	39.13 ± 0.58
	SAR-EM	47.79 ± 0.31	49.54 ± 0.14	50.27 ± 0.43	60.80 ± 0.21	60.30 ± 0.32	44.90 ± 0.73
	A^3	80.21 ± 1.07 –	74.89 ± 1.62 ✓	82.33 ± 1.49 –	**90.73 ± 0.27** ✓	**79.34 ± 0.69** ✓	**89.84 ± 1.57** ✓
	IsolationForest	80.63 ± 1.16 –	75.63 ± 1.54 ✓	**87.39 ± 2.17** ✓	90.08 ± 0.47 –	75.64 ± 0.45 ✓	80.17 ± 3.24 ✓
	ECODv2	80.44 ± 1.08 –	73.83 ± 1.44 ✓	80.05 ± 1.68 –	90.39 ± 0.27 ✓	75.17 ± 0.60 ✓	80.16 ± 2.98 ✓
	OC-SVM	**80.73 ± 1.23** –	**75.70 ± 1.58** ✓	80.33 ± 1.71 –	90.39 ± 0.27 ✓	75.19 ± 0.60 ✓	81.14 ± 3.14 ✓
0.10	Baseline	85.11 ± 0.87	74.24 ± 1.59	87.70 ± 1.07	82.38 ± 2.32	81.86 ± 0.82	61.63 ± 0.75
	Baseline (orig)	81.44 ± 0.57	65.01 ± 0.84	85.67 ± 0.96	81.70 ± 2.68	81.82 ± 0.86	61.70 ± 0.76
	LBE	51.54 ± 0.35	51.48 ± 0.17	50.93 ± 0.38	62.47 ± 2.10	60.73 ± 0.29	39.30 ± 0.58
	SAR-EM	51.25 ± 0.31	49.36 ± 0.13	52.58 ± 0.40	65.27 ± 0.40	61.82 ± 0.27	47.64 ± 1.12
	A^3	90.01 ± 0.47 ✓	83.14 ± 1.41 ✓	89.37 ± 0.53 ✓	**92.35 ± 0.28** ✓	**83.31 ± 0.33** –	**88.42 ± 1.78** ✓
	IsolationForest	90.52 ± 0.41 ✓	**83.60 ± 1.28** ✓	**90.80 ± 0.38** ✓	90.21 ± 0.57 ✓	83.22 ± 0.28 –	62.84 ± 0.95 –
	ECODv2	**90.82 ± 0.37** ✓	81.97 ± 1.30 ✓	89.90 ± 0.31 ✓	92.09 ± 0.30 ✓	83.10 ± 0.32 –	66.45 ± 1.89 ✓
	OC-SVM	90.75 ± 0.43 ✓	83.57 ± 1.28 ✓	89.89 ± 0.29 ✓	92.10 ± 0.30 ✓	83.17 ± 0.31 –	63.66 ± 1.04 –
0.30	Baseline	84.50 ± 0.50	76.38 ± 1.56	86.09 ± 1.47	75.17 ± 1.97	81.73 ± 0.53	60.98 ± 0.57
	Baseline (orig)	85.57 ± 0.59	72.03 ± 0.65	82.71 ± 1.29	80.76 ± 2.04	81.22 ± 0.95	61.00 ± 0.57
	LBE	62.72 ± 0.44	58.58 ± 0.84	80.62 ± 4.85	73.07 ± 2.05	69.93 ± 3.10	81.32 ± 7.27
	SAR-EM	60.85 ± 0.27	52.09 ± 0.19	64.83 ± 0.39	76.37 ± 0.49	70.43 ± 0.38	66.94 ± 1.86
	A^3	92.47 ± 0.32 ✓	90.49 ± 0.23 ✓	89.87 ± 0.65 ✓	**93.45 ± 0.12** ✓	85.16 ± 0.41 ✓	**90.30 ± 2.59** ✓
	IsolationForest	92.68 ± 0.31 ✓	90.59 ± 0.23 ✓	**92.67 ± 0.25** ✓	92.02 ± 0.44 ✓	85.13 ± 0.36 ✓	72.12 ± 3.10 ✓
	ECODv2	92.50 ± 0.32 ✓	89.55 ± 0.21 ✓	90.66 ± 0.35 ✓	93.08 ± 0.12 ✓	85.08 ± 0.35 ✓	77.92 ± 2.90 ✓
	OC-SVM	**92.71 ± 0.31** ✓	**90.67 ± 0.22** ✓	90.67 ± 0.37 ✓	93.08 ± 0.12 ✓	**85.22 ± 0.34** ✓	74.90 ± 1.97 ✓
0.50	Baseline	86.37 ± 0.59	80.51 ± 0.99	87.44 ± 0.90	80.42 ± 2.71	81.39 ± 0.36	66.74 ± 3.03
	Baseline (orig)	88.74 ± 0.58	80.40 ± 0.82	83.56 ± 1.14	77.91 ± 1.81	81.97 ± 0.71	61.32 ± 0.61
	LBE	72.72 ± 0.43	66.42 ± 1.45	90.34 ± 1.11	79.91 ± 4.99	82.60 ± 2.32	**93.84 ± 2.73**
	SAR-EM	70.51 ± 0.24	59.13 ± 0.20	81.96 ± 0.47	87.94 ± 0.40	83.37 ± 0.37	83.52 ± 1.49
	A^3	92.75 ± 1.11 ✓	92.24 ± 0.25 ✓	92.23 ± 0.32 ✓	**93.44 ± 0.12** ✓	**87.00 ± 0.35** ✓	83.15 ± 3.82 ✓
	IsolationForest	92.92 ± 1.02 ✓	92.72 ± 0.22 ✓	**93.50 ± 0.21** ✓	92.49 ± 0.30 ✓	86.92 ± 0.37 ✓	71.49 ± 3.58 ✓
	ECODv2	**92.93 ± 1.04** ✓	91.97 ± 0.23 ✓	92.31 ± 0.19 ✓	93.41 ± 0.15 ✓	86.85 ± 0.36 ✓	81.05 ± 2.42 ✓
	OC-SVM	92.80 ± 1.05 ✓	**92.79 ± 0.21** ✓	92.90 ± 0.24 ✓	93.40 ± 0.14 ✓	86.98 ± 0.36 ✓	70.27 ± 3.19 ✓
0.70	Baseline	90.20 ± 0.68	85.61 ± 1.08	87.01 ± 0.65	85.01 ± 1.44	83.73 ± 0.29	61.17 ± 0.53
	Baseline (orig)	90.55 ± 0.52	87.87 ± 0.66	87.74 ± 1.19	87.14 ± 1.42	84.87 ± 0.52	61.22 ± 0.53
	LBE	82.33 ± 0.50	66.56 ± 3.24	91.49 ± 1.27	87.97 ± 3.83	83.13 ± 2.55	**98.10 ± 0.46**
	SAR-EM	80.45 ± 0.21	79.92 ± 0.13	92.66 ± 0.19	**94.14 ± 0.13**	**88.77 ± 0.30**	92.92 ± 0.63
	A^3	93.54 ± 0.79 ✓	94.09 ± 0.31 ✓	93.21 ± 0.23 ✓	93.99 ± 0.07 ✓	88.31 ± 0.33 ✓	95.13 ± 0.70 ✓
	IsolationForest	94.02 ± 0.62 ✓	94.36 ± 0.27 ✓	**93.62 ± 0.20** ✓	93.74 ± 0.10 ✓	88.36 ± 0.36 ✓	72.28 ± 2.39 ✓
	ECODv2	93.55 ± 0.73 ✓	93.70 ± 0.29 ✓	93.44 ± 0.23 ✓	93.81 ± 0.09 ✓	88.51 ± 0.28 ✓	94.22 ± 1.03 ✓
	OC-SVM	**94.05 ± 0.66** ✓	**94.39 ± 0.28** ✓	93.47 ± 0.23 ✓	93.77 ± 0.09 ✓	88.44 ± 0.33 ✓	91.95 ± 0.98 ✓

Table 4.3: F1 score values per dataset. Green ticks („✓”) correspond to the cases when t-test rejected equality of F1 scores of the VAE-PU-OCC method considered and that of the baseline original method in favor of the former one at $\alpha = 0.05$. Dashes („—”) indicate the failure to reject (see Section 4.8.3). Bold entries correspond to maximum mean F1 score for a given dataset and label frequency combination.

c	Method	MNIST	MNIST	CIFAR	CIFAR	STL	Gas
		3v5	OvE	CarTruck	MachineAnimal	MachineAnimal	Concentrations
0.02	Baseline	**82.59 ± 0.85**	75.85 ± 0.77	75.01 ± 4.94	86.11 ± 0.87	67.54 ± 1.73	83.62 ± 1.60
	Baseline (orig)	80.91 ± 0.85	68.48 ± 0.50	77.23 ± 3.82	86.08 ± 1.38	68.14 ± 1.45	81.06 ± 1.79
	LBE	3.64 ± 0.37	2.14 ± 0.20	0.81 ± 0.19	0.32 ± 0.07	1.31 ± 0.34	0.21 ± 0.08
	SAR-EM	2.96 ± 0.28	1.73 ± 0.09	2.19 ± 0.50	3.51 ± 0.33	2.91 ± 0.37	14.56 ± 1.23
	A^3	82.56 ± 0.86 —	76.97 ± 1.17 ✓	82.75 ± 1.31 —	**88.96 ± 0.30** ✓	**75.24 ± 0.62** ✓	**91.73 ± 1.37** ✓
	IsolationForest	82.53 ± 0.92 —	77.26 ± 1.12 ✓	**87.72 ± 2.01** ✓	88.22 ± 0.47 —	70.92 ± 0.65 ✓	84.81 ± 2.06 —
	ECODv2	**82.59 ± 0.92** —	76.50 ± 0.98 ✓	79.62 ± 1.81 —	88.57 ± 0.29 ✓	70.49 ± 0.68 —	84.96 ± 1.69 —
	OC-SVM	82.54 ± 0.96 —	**77.27 ± 1.11** ✓	79.73 ± 1.89 —	88.57 ± 0.29 ✓	70.49 ± 0.67 —	85.96 ± 1.79 ✓
0.10	Baseline	87.50 ± 0.63	79.17 ± 1.04	88.94 ± 0.79	82.11 ± 1.92	79.89 ± 0.55	76.04 ± 0.52
	Baseline (orig)	84.26 ± 0.47	72.43 ± 0.30	87.34 ± 0.70	81.62 ± 2.22	79.61 ± 0.49	76.08 ± 0.53
	LBE	17.18 ± 0.74	12.11 ± 0.71	4.78 ± 0.70	8.97 ± 6.99	4.70 ± 0.93	0.77 ± 0.20
	SAR-EM	16.13 ± 0.43	9.60 ± 0.20	11.70 ± 0.76	24.02 ± 1.22	9.98 ± 0.56	22.71 ± 2.98
	A^3	90.65 ± 0.39 ✓	83.95 ± 1.13 ✓	90.18 ± 0.43 ✓	**90.81 ± 0.32** ✓	**80.66 ± 0.28** ✓	**91.31 ± 1.14** ✓
	IsolationForest	91.12 ± 0.36 ✓	84.32 ± 1.08 ✓	**91.31 ± 0.36** ✓	88.65 ± 0.55 ✓	80.60 ± 0.25 ✓	76.52 ± 0.60 —
	ECODv2	**91.34 ± 0.32** ✓	83.20 ± 1.06 ✓	90.59 ± 0.28 ✓	90.51 ± 0.33 ✓	80.53 ± 0.27 —	78.33 ± 1.02 ✓
	OC-SVM	91.30 ± 0.37 ✓	**84.33 ± 1.07** ✓	90.58 ± 0.25 ✓	90.51 ± 0.33 ✓	80.56 ± 0.26 —	76.93 ± 0.65 —
0.30	Baseline	87.16 ± 0.36	80.99 ± 1.05	87.66 ± 1.14	76.34 ± 1.52	80.57 ± 0.43	75.73 ± 0.45
	Baseline (orig)	87.64 ± 0.44	77.02 ± 0.30	85.15 ± 0.96	80.68 ± 1.60	80.17 ± 0.66	75.74 ± 0.45
	LBE	47.99 ± 0.75	40.57 ± 1.00	71.66 ± 9.77	56.89 ± 7.47	53.68 ± 8.28	74.83 ± 11.37
	SAR-EM	43.02 ± 0.43	30.21 ± 0.29	47.12 ± 0.85	59.32 ± 1.19	43.02 ± 0.79	61.74 ± 3.13
	A^3	92.84 ± 0.29 ✓	90.65 ± 0.23 ✓	90.51 ± 0.55 ✓	**91.97 ± 0.12** ✓	82.74 ± 0.36 ✓	**92.86 ± 1.57** ✓
	IsolationForest	93.07 ± 0.28 ✓	90.78 ± 0.22 ✓	**92.70 ± 0.25** ✓	90.46 ± 0.43 ✓	82.67 ± 0.35 ✓	81.63 ± 1.80 ✓
	ECODv2	92.91 ± 0.28 ✓	89.92 ± 0.20 ✓	91.15 ± 0.33 ✓	91.50 ± 0.13 ✓	82.69 ± 0.33 ✓	84.88 ± 1.65 ✓
	OC-SVM	**93.10 ± 0.27** ✓	**90.89 ± 0.20** ✓	91.17 ± 0.35 ✓	91.50 ± 0.13 ✓	**82.78 ± 0.31** ✓	82.95 ± 1.12 ✓
0.50	Baseline	88.47 ± 0.49	83.90 ± 0.68	88.70 ± 0.72	80.81 ± 2.18	80.61 ± 0.27	78.12 ± 1.53
	Baseline (orig)	90.08 ± 0.47	83.11 ± 0.53	85.78 ± 0.88	78.39 ± 1.39	81.05 ± 0.58	75.89 ± 0.46
	LBE	67.41 ± 0.58	64.63 ± 1.53	90.81 ± 0.93	79.75 ± 3.31	80.05 ± 1.77	**94.27 ± 2.71**
	SAR-EM	62.77 ± 0.19	51.08 ± 0.26	78.75 ± 0.71	82.96 ± 0.64	75.04 ± 0.61	84.07 ± 1.74
	A^3	93.29 ± 0.95 ✓	92.49 ± 0.24 ✓	92.47 ± 0.27 ✓	**92.06 ± 0.13** ✓	**84.42 ± 0.39** ✓	87.88 ± 2.30 ✓
	IsolationForest	**93.42 ± 0.89** ✓	92.90 ± 0.21 ✓	**93.46 ± 0.20** ✓	91.11 ± 0.29 ✓	84.37 ± 0.32 ✓	80.91 ± 1.97 ✓
	ECODv2	93.39 ± 0.94 ✓	92.20 ± 0.22 ✓	92.54 ± 0.21 ✓	91.96 ± 0.16 ✓	84.31 ± 0.32 ✓	86.18 ± 1.57 ✓
	OC-SVM	93.31 ± 0.91 ✓	**92.96 ± 0.20** ✓	93.09 ± 0.22 ✓	91.95 ± 0.15 ✓	**84.42 ± 0.32** ✓	80.21 ± 1.64 ✓
0.70	Baseline	91.40 ± 0.57	87.54 ± 0.83	88.35 ± 0.52	84.14 ± 1.27	82.47 ± 0.25	75.82 ± 0.42
	Baseline (orig)	91.41 ± 0.44	89.04 ± 0.51	89.00 ± 0.95	86.08 ± 1.27	83.41 ± 0.43	75.85 ± 0.42
	LBE	82.33 ± 0.45	73.48 ± 1.62	91.92 ± 1.07	87.50 ± 3.08	81.71 ± 2.03	**98.42 ± 0.38**
	SAR-EM	78.42 ± 0.21	77.94 ± 0.16	92.42 ± 0.23	92.53 ± 0.16	84.98 ± 0.38	93.79 ± 0.59
	A^3	93.96 ± 0.71 ✓	94.20 ± 0.29 ✓	93.31 ± 0.24 ✓	**92.58 ± 0.09** ✓	85.84 ± 0.28 ✓	96.13 ± 0.54 ✓
	IsolationForest	94.33 ± 0.58 ✓	94.46 ± 0.26 ✓	**93.62 ± 0.22** ✓	92.28 ± 0.10 ✓	**85.93 ± 0.35** ✓	81.53 ± 1.51 ✓
	ECODv2	93.95 ± 0.68 ✓	93.86 ± 0.27 ✓	93.51 ± 0.25 ✓	92.35 ± 0.11 ✓	**85.93 ± 0.26** ✓	95.50 ± 0.77 ✓
	OC-SVM	**94.38 ± 0.62** ✓	**94.49 ± 0.27** ✓	93.56 ± 0.24 ✓	92.30 ± 0.11 ✓	**85.93 ± 0.27** ✓	93.83 ± 0.69 ✓

high enough c values) than its competitors. In other test cases, however, its performance is relatively subpar – similarly to SAR-EM, its F1 score plummets when decreasing the proportion of labeled positive examples. This establishes low label frequency performance as a substantial advantage of generative models over the algorithms which model the propensity score explicitly such as SAR-EM and LBE. We investigate the possible causes of this behavior further on in this section.

VAE-PU+OCC achieves excellent classification results on all of the benchmark datasets, as measured by the accuracy and F1 score. All of the proposed OCC variants outperform baseline VAE-PU models in all of the test cases. In some cases performance increase is very slight (e.g. MNIST 3v5, $c = 0.02$), where only a fraction of a percent increase in accuracy is observed; but there are also cases where classification performance rises dramatically (see Gas Concentration results), up to tens of percentage points (pp.). Overall, applying the OCC variants results in a substantial increase in both the accuracy and F1 score, in most cases by a several pp. Difference between different VAE-PU+OCC methods presented in the tables are usually slight (generally below 1 pp. for both accuracy and F1 score). Nevertheless, there are also scenarios where one of the methods performs significantly better that the other – A^3 dominates competitors on Gas Concentrations dataset, while Isolation Forest performs significantly better for CIFAR CarTruck data. Overall, these two (Isolation Forest and A^3) variants are the most noteworthy – they outperform competitors on multiple datasets, while remaining competitive in scenarios which proved more difficult for the methods. Closer look at the results reveals that competing methods (VAE-PU, SAR-EM and LBE) tend to introduce precision-recall imbalance (for VAE-PU, the recall is often much higher of the two, while SAR-EM and LBE are skewed towards the precision; for detailed metric values and discussion, refer to Section 4.8.2), while OCC variants of VAE-PU achieve balanced results in nearly all test cases.

Another significant feature of the VAE-PU+OCC variants is stabilization of the results. Note that the standard error of the mean (SEM) for all proposed methods decreases significantly (as compared to the baseline VAE-PU) in a majority of test cases. Notable exception is the Gas Concentration dataset, where even though SEM usually increases, it occurs with a simultaneous significant classification performance improvement. VAE-PU+OCC also offers drastically lowered training time (up to 10 times shorter, compared to alternatives such as SAR-EM and LBE; detailed time values will be presented in Section 4.8.5), which makes it attractive even in rare cases where its accuracy is lower. Naturally, it is also slower than baseline VAE-PU, but this loss doesn't usually exceed 20% extra training time.

Even though due to already long time required to obtain experimental results we limited the number of tested OCC methods to four, we feel that it is a representative sample of approaches, incorporating both classic and modern models and ranging from simple, statistical methods to neural-network based classifiers. As a result of the experiments we suggest that A^3-based variant of VAE-PU+OCC to be recommended in most practical scenarios, due to exceptional performance in several scenarios while maintaining strong baseline accuracy in general case.

4.8.2 Precision-recall balance

Table 4.4 presents detailed experimental results on MNIST 3v5 dataset, which reveals interesting details concerning the precision and the recall for each of the tested methods. Note that in case of SAR-EM and LBE, the precision is really high and stays almost constant, regardless of label frequency; on the other hand, for low c values the recall of the methods is extremely poor, indicating far too small number of positive predictions. Conversely, VAE-PU (in both versions) tends to skew the precision-recall balance the other way – even though the recall of the method is really remarkable, high number of False Positives means that both the precision and the overall predictive power of the model suffer. OCC-based example selection helps VAE-PU achieve equilibrium of the precision and the recall. This effect is more noticeable for higher label frequencies – in all cases we can see a significant precision increase at the expense of slight recall decrease, but starting at $c = 0.10$ it causes precision and recall values to be nearly equal. Such behavior is consistent for all of the benchmark datasets, therefore the precision and the recall values were reported only in this case to avoid visual clutter in the tables. Relatively small precision values for low label frequencies might be explained by the imperfections of generated examples – when portion of labeled positives is small, it is harder to learn a robust positive data representation, as the labeled set is poorly representative of the positive distribution due to biased PU problem nature. The generative process itself also emphasizes this issue – small labeled set causes the generated examples to be limited in diversity, as discussed at the end of Chapter 4.6.2.

4.8.3 T-test p-values

The tables 4.5 and 4.6 contain p-values of t-test of the null hypothesis that the accuracy or, respectively, F1 measure of the specific VAE-PU-OCC variant is equal to that of the baseline original method, against the alternative that it is larger than the baseline. Green ticks correspond to the cases when the null is rejected at $\alpha = 0.05$ significance level and dashes signify the failure to reject. As the averages of the metrics are based on 10 repetitions of the experiment, the benchmark distribution under the null hypothesis is t distribution with $10 + 10 - 2 = 18$ degrees of freedom.

Table 4.4: MNIST 3v5 results – no-SCAR.

c	Method	Accuracy	Precision	Recall	F1 score
0.02	Baseline	79.99 ± 1.04	77.27 ± 0.99	**88.75 ± 0.85**	**82.59 ± 0.85**
	Baseline (orig)	78.18 ± 0.97	75.65 ± 0.98	87.06 ± 1.17	80.91 ± 0.85
	LBE	47.78 ± 0.29	**94.95 ± 1.75**	1.86 ± 0.19	3.64 ± 0.37
	SAR-EM	47.79 ± 0.31	94.33 ± 1.63	1.51 ± 0.14	2.96 ± 0.28
	A^3	80.21 ± 1.07	78.20 ± 1.10	87.50 ± 0.82	82.56 ± 0.86
	ECODv2	80.44 ± 1.08	78.53 ± 1.23	87.18 ± 0.86	**82.59 ± 0.92**
	IsolationForest	80.63 ± 1.16	80.04 ± 1.45	85.37 ± 1.01	82.53 ± 0.92
	OC-SVM	**80.73 ± 1.23**	80.57 ± 1.68	84.86 ± 1.01	82.54 ± 0.96
0.10	Baseline	85.11 ± 0.87	79.35 ± 1.06	**97.61 ± 0.27**	87.50 ± 0.63
	Baseline (orig)	81.44 ± 0.57	76.80 ± 0.72	93.42 ± 0.84	84.26 ± 0.47
	LBE	51.54 ± 0.35	93.77 ± 0.63	9.47 ± 0.45	17.18 ± 0.74
	SAR-EM	51.25 ± 0.31	**94.45 ± 0.66**	8.82 ± 0.26	16.13 ± 0.43
	A^3	90.01 ± 0.47	90.45 ± 0.69	90.88 ± 0.41	90.65 ± 0.39
	ECODv2	**90.82 ± 0.37**	91.69 ± 0.61	91.06 ± 0.72	**91.34 ± 0.32**
	IsolationForest	90.52 ± 0.41	90.83 ± 0.54	91.44 ± 0.56	91.12 ± 0.36
	OC-SVM	90.75 ± 0.43	91.42 ± 0.62	91.23 ± 0.54	91.30 ± 0.37
0.30	Baseline	84.50 ± 0.50	77.99 ± 0.64	**98.83 ± 0.15**	87.16 ± 0.36
	Baseline (orig)	85.57 ± 0.59	80.66 ± 0.88	96.01 ± 0.30	87.64 ± 0.44
	LBE	62.72 ± 0.44	92.85 ± 0.49	32.39 ± 0.69	47.99 ± 0.75
	SAR-EM	60.85 ± 0.27	**95.09 ± 0.47**	27.81 ± 0.37	43.02 ± 0.43
	A^3	92.47 ± 0.32	93.85 ± 0.42	91.89 ± 0.55	92.84 ± 0.29
	ECODv2	92.50 ± 0.32	93.61 ± 0.57	92.25 ± 0.50	92.91 ± 0.28
	IsolationForest	92.68 ± 0.31	93.87 ± 0.51	92.31 ± 0.47	93.07 ± 0.28
	OC-SVM	**92.71 ± 0.31**	93.76 ± 0.47	92.48 ± 0.46	**93.10 ± 0.27**
0.50	Baseline	86.37 ± 0.59	80.47 ± 0.42	**98.25 ± 0.65**	88.47 ± 0.49
	Baseline (orig)	88.74 ± 0.58	84.87 ± 0.74	96.01 ± 0.26	90.08 ± 0.47
	LBE	72.72 ± 0.43	92.31 ± 0.55	53.13 ± 0.74	67.41 ± 0.58
	SAR-EM	70.51 ± 0.24	**95.48 ± 0.31**	46.76 ± 0.25	62.77 ± 0.19
	A^3	92.75 ± 1.11	92.72 ± 1.32	93.91 ± 0.65	93.29 ± 0.95
	ECODv2	**92.93 ± 1.04**	93.21 ± 1.19	93.61 ± 0.91	93.39 ± 0.94
	IsolationForest	92.92 ± 1.02	93.07 ± 1.17	93.81 ± 0.74	**93.42 ± 0.89**
	OC-SVM	92.80 ± 1.05	93.03 ± 1.23	93.62 ± 0.66	93.31 ± 0.91
0.70	Baseline	90.20 ± 0.68	85.78 ± 0.74	**97.85 ± 0.55**	91.40 ± 0.57
	Baseline (orig)	90.55 ± 0.52	88.58 ± 0.71	94.46 ± 0.46	91.41 ± 0.44
	LBE	82.33 ± 0.50	88.17 ± 1.21	77.47 ± 1.21	82.33 ± 0.45
	SAR-EM	80.45 ± 0.21	94.92 ± 0.35	66.81 ± 0.23	78.42 ± 0.21
	A^3	93.54 ± 0.79	93.73 ± 0.83	94.22 ± 0.72	93.96 ± 0.71
	ECODv2	93.55 ± 0.73	93.86 ± 0.75	94.08 ± 0.88	93.95 ± 0.68
	IsolationForest	94.02 ± 0.62	**95.06 ± 0.46**	93.65 ± 0.89	94.33 ± 0.58
	OC-SVM	**94.05 ± 0.66**	94.81 ± 0.47	93.97 ± 0.87	**94.38 ± 0.62**

Table 4.5: T-test p-values for Accuracy per dataset.

c	Method	MNIST	MNIST	CIFAR	CIFAR	STL	Gas
		3v5	OvE	CarTruck	MachineAnimal	MachineAnimal	Concentrations
0.02	A^3	0.09 —	< 0.01 ✓	0.16 —	0.03 ✓	< 0.01 ✓	< 0.01 ✓
	IsolationForest	0.06 —	< 0.01 ✓	0.01 ✓	0.07 —	< 0.01 ✓	0.03 ✓
	ECODv2	0.07 —	< 0.01 ✓	0.41 —	0.04 ✓	0.02 ✓	0.03 ✓
	OC-SVM	0.06 —	< 0.01 ✓	0.38 —	0.04 ✓	0.02 ✓	0.02 ✓
0.10	A^3	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓	0.06 —	< 0.01 ✓
	IsolationForest	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓	0.07 —	0.18 —
	ECODv2	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓	0.09 —	0.02 ✓
	OC-SVM	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓	0.08 —	0.07 —
0.30	A^3	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓
	IsolationForest	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓
	ECODv2	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓
	OC-SVM	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓
0.50	A^3	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓
	IsolationForest	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓	0.01 ✓
	ECODv2	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓
	OC-SVM	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓	0.01 ✓
0.70	A^3	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓
	IsolationForest	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓
	ECODv2	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓
	OC-SVM	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓

Table 4.6: T-test p-values for F1 score per dataset.

c	Method	MNIST	MNIST	CIFAR	CIFAR	STL	Gas
		3v5	OvE	CarTruck	MachineAnimal	MachineAnimal	Concentrations
0.02	A^3	0.09 —	< 0.01 ✓	0.09 —	0.03 ✓	< 0.01 ✓	< 0.01 ✓
	IsolationForest	0.11 —	< 0.01 ✓	0.01 ✓	0.08 —	0.05 ✓	0.09 —
	ECODv2	0.10 —	< 0.01 ✓	0.29 —	0.05 ✓	0.08 —	0.07 —
	OC-SVM	0.11 —	< 0.01 ✓	0.28 —	0.05 ✓	0.08 —	0.03 ✓
0.10	A^3	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓	0.04 ✓	< 0.01 ✓
	IsolationForest	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓	0.04 ✓	0.29 —
	ECODv2	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓	0.06 —	0.03 ✓
	OC-SVM	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓	0.05 —	0.16 —
0.30	A^3	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓
	IsolationForest	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓
	ECODv2	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓
	OC-SVM	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓
0.50	A^3	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓
	IsolationForest	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓	0.01 ✓
	ECODv2	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓
	OC-SVM	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓	0.01 ✓
0.70	A^3	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓
	IsolationForest	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓
	ECODv2	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓
	OC-SVM	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓	< 0.01 ✓

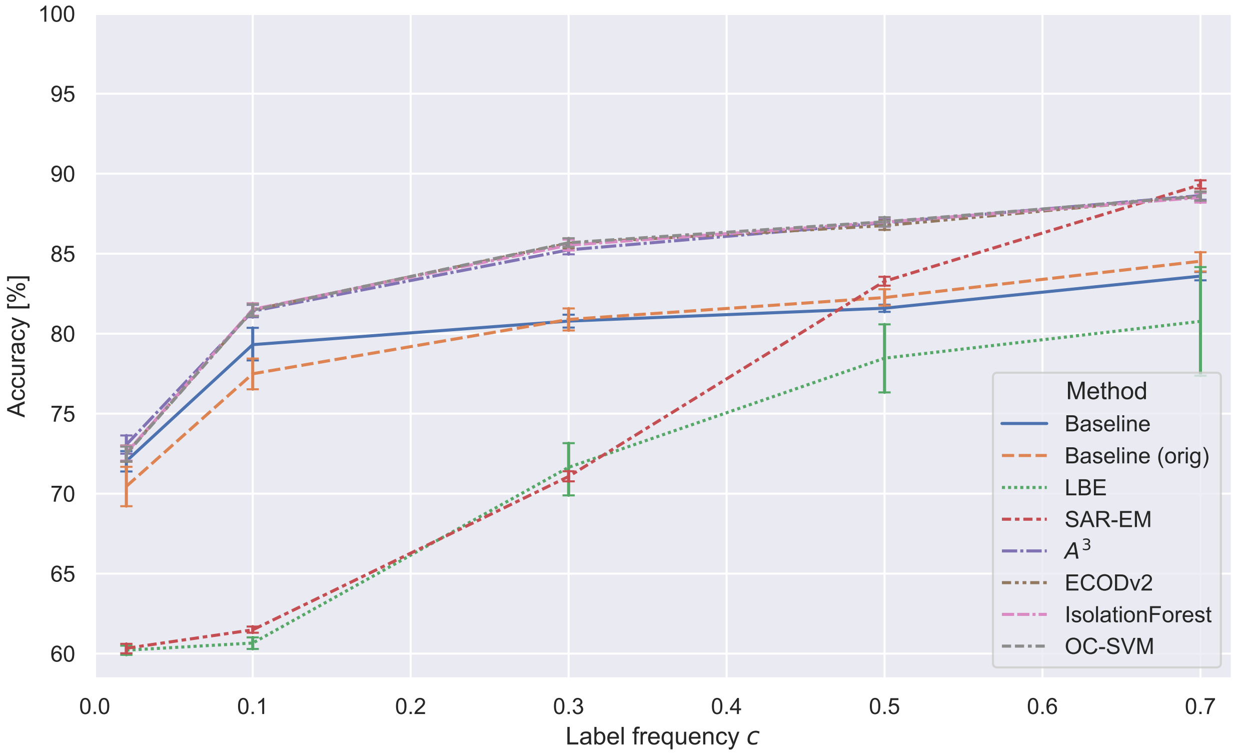


Figure 4.9: Test accuracy in SCAR setting, STL MachineAnimal dataset.

4.8.4 Performance in SCAR setting

As the presented approach does not make any assumptions on the nature of biased sampling it is conjectured that it will also perform well under SCAR. In order to verify the hypothesis that OCC variants perform well even in SCAR setting, despite more general assumptions, we performed additional tests on STL dataset for this scenario. Here, items in the training set are labeled according to the SCAR assumption, i.e. probability of being labeled for each positive example (propensity score) is constant and equal to label frequency c. As SCAR scenario is a special case of no-SCAR PU learning, it is expected that the results will stay similar to no-SCAR experiments.

Figure 4.9 illustrates results for the experiments in SCAR setting. The initial assumptions are confirmed – for all label frequencies, OCC variants outperform the baseline VAE-PU models, whereas SAR-EM performs poorly for low c values, while being competitive or even slightly outperforming the competition in high label frequency setting; LBE was outperformed in all test cases. Note that in many use cases the minor increase in classification performance of SAR-EM might be outweighed by a severe training time increase; nevertheless, some critical applications might find the high computational cost feasible. Also, performance of various OCC variants is almost indistinguishable in this case. Overall, the results follow similar patterns to no-SCAR

Table 4.7: Training time per dataset ($c = 0.5$).

Method	**MNIST**	**MNIST**	**CIFAR**	**CIFAR**	**STL**	**Gas**
	3v5	OvE	CarTruck	MachineAnimal	MachineAnimal	Concentrations
Baseline (modified)	258.77s	1617.03s	239.76s	1527.38s	255.23s	**74.90s**
Baseline (original)	**244.85s**	**1272.71s**	**215.28s**	**1037.12s**	**238.16s**	77.25s
SAR-EM	873.65s	44142.47s	9232.45s	47772.81s	5278.49s	87.37s
LBE	668.86s	12958.04s	927.65s	8984.34s	1090.33s	697.30s
VAE-PU+A^3	282.06s	1797.25s	256.86s	1635.81s	272.72s	87.80s
VAE-PU+ECOD	285.92s	1799.67s	256.44s	1645.48s	273.10s	82.40s
VAE-PU+IsolationForest	289.93s	1935.04s	264.96s	1673.16s	273.69s	86.90s
VAE-PU+OC-SVM	369.05s	3334.41s	249.64s	1798.46s	275.99s	79.30s

scenario, which proves the effectiveness of our method even in a SCAR setting, demonstrating the robustness of the VAE-PU+OCC approach.

4.8.5 Training time comparison

Table 4.7 contains information about typical training times for a given algorithm on each dataset. Modified VAE-PU model tends to train slower than the original version, and to no surprise training time increases when using VAE-PU+OCC as opposed to the baseline; it should be emphasized, though, that the extra training step does not come at a heavy computational cost, as the typical training time increase ranges from 10% to 20%. A significant exception is One-Class SVM variant, which can be slow for large datasets – this can be seen especially for MNIST OvE dataset, where the training time doubled after OC-SVM training. This example also highlights a significant advantage of VAE-PU-based models over the SAR-EM and LBE algorithm; when training dataset is not small, SAR-EM can reach training times up to 40-50 times larger than the other competitors, while LBE training is still comparably long even for small datasets. This property makes using VAE-PU and VAE-PU+OCC significantly more attractive even in high label frequency settings, where they offer way lower computational time combined with classification accuracy is on par or better than that of SAR-EM and LBE.

4.9 Conclusions

VAE-PU+OCC builds upon an innovative VAE-PU model, which proved the strength of generative approaches in no-SCAR PU data modeling. Through the application of one-class classification methods, both modern and traditional, the extended model has shown excellent results in a diverse array of experiments. The highlight is the outstanding accuracy of the models in medium label frequency settings – for low label frequencies, the gains of OCC-based models are minor relative to VAE-PU baselines, whereas for high label frequencies classical, non-generative algorithms such as SAR-EM and LBE remain competitive. This chapter proves that application of one-class classification techniques in no-SCAR PU learning provides a substantial improvement.

Chapter 5

False Omission Rate control

5.1 Introduction

In the previous chapter we considered detection of negatives among unlabeled observations based on scores of an outlier detector and expected number of such elements in unlabeled set. In the following we consider an alternative approach to construct cutoff-point, which is based on hypothesis testing method which aims at control of ratio of negative observations among those considered positive unlabeled. The rationale here is to ensure sufficient purity of the set of observations deemed positive. We study first a general problem of False Omission Rate control.

We consider the situation when a score statistic is learned on a random sample of regular observations (inliers) and used to detect out of distribution observations (outliers) with an objective to control the percentage of undetected outliers among the observations classified as inliers. Such a need arises in many practical situations: imagine a scrutiny of possibly fraudulent transactions for which one would like to detect all but a very small percent of frauds – such an approach accounts for the fact that trying to detect all frauds will require very stringent safety rules which would deter potential customers. Another example is development of a new test for a contagious disease (e.g. COVID-19), for which is vital to ensure that randomly chosen person will not pass it *if infected* with large probability (see Takahashi, Ichinose, and Yasusei (2022)). In such situations it is much more important to control False Omission Rate (FOR, called also False Non-discovery Rate (FNR) see Efron (2010)) than commonly used False Discovery Rate (FDR). FOR is defined as the expected value of False Omission Proportion i.e. proportion of undetected outliers among observations classified as inliers, whereas FDR is the expected proportion of inliers among observations deemed outliers (Harvey and Liu 2017; Genovese and Wasserman 2002). Obviously, in many situations FDR control has evident advantages, but we argue that in numerous cases FOR control – or its Bayesian analogue defined below – is of main interest, and procedures which ensure it are worth studying. Note that for scenario of testing a single hypothesis, this corresponds to controlling the error of type II (rather than the the type I error, as in the case of FDR).

Our main objective here is to develop a rule which approximately controls FOR and to investigate its properties both theoretically and by means of analysis of real data sets, and then apply it to PU data. The rule developed here is derived analogously to Benjamini-Hochberg rule (Benjamini and Hochberg 1995) using Frequentist Bayes approach (see e.g. Efron (2010), Chapter 4). We also consider several methods scores for outlier detection and check how their choice influences control of FOR. We then investigate ways of diminishing intrinsic variability of p-values due to the random split of the data set, and evaluate the FOR control procedure in the outlier detection setting. Finally, we consider the application of the procedure back in the VAE-PU framework, where the FOR control procedure is used to select the best inliers (positive examples) in unlabeled set, thus improving the purity of the resultant positive-unlabeled set. Thus, we still tackle a central problem from the last chapter – discovery of positive unlabeled observations among unlabeled ones – but this time, looking at it from the perspective of multiple hypothesis testing in outlier detection problem.

5.2 FOR control procedure

5.2.1 Preliminaries

Consider checking whether observations under study are outliers with the use of a specific score statistic $\widehat{s}$ to test a null hypothesis

$$H_{0,i} : \text{the observation is an inlier}$$

versus an alternative

$$H_{1,i} : \text{the observation is an outlier.}$$

We adopt throughout the convention that large values of $\widehat{s}$ indicate outliers. It is known that when the cumulative distribution function (CDF) denoted by F of a test statistic is continuous, the distribution of the corresponding p-value equal to $1 - F(X_i)$, provided the null hypothesis is true, is uniform on $[0,1]$ (Lehmann and Romano 2022). This is the fundamental property used to bound Family Wise Error Rate (FWER) defined as probability of falsely rejecting at least one null, or False Discovery Rate (FDR) defined below, which is more easily controlled. For a discussion of numerous solutions to the problem from which Benjamini-Hochberg (BH) procedure is the most commonly used, see e.g. Dudoit and Laan (2008). In Genovese and Wasserman (2002) analysis of behavior of FOR for BH procedure is given. However, construction of rules controlling FOR remains, to the best of our knowledge, largely untreated. Imagine now that we have a set of n observations generated by mixture of distribution of inliers (occurring with probability π) and outliers (occurring with probability $1-\pi$), and denote by $p_1, \ldots, p_n$ corresponding p-values of a test under consideration. Then we can write

$$p_i \sim \pi U + (1-\pi)F_1, \quad i = 1, \ldots, n, \tag{5.1}$$

where U stands for the distribution function of the uniform distribution $U[0,1]$: $U(t) = t$, F_1 is the cumulative distribution function of the p-values for outliers and "$\sim$" denotes "is distributed as". In the following we assume that mixing proportion π is known. This assumption is commonly met i.e. when prevalence of a certain disease can be precisely estimated based on independent data base. We also assume that p_i are independent random variables.

We suppose that n_0 observations are inliers (nulls) and $n_1 = n - n_0$ are outliers (non-nulls) and note that due to our mixture assumption (5.1) n_0 and n_1 are random and have Bernoulli distribution: $n_0 \sim \text{Bin}(n, \pi)$ and $n_1 \sim \text{Bin}(n, 1-\pi)$. Consider a specific decision rule assigning each of n observations to inliers or outliers and denote by R the number of of rejected null hypotheses, by V the number of falsely rejected nulls and by Z the number of falsely not rejected alternatives. Note that $Z = n_1 - (R - V)$ (as $R - V$ is the number of rejected outliers) and let NR be the number of not rejected items. We will consider threshold rules such that for any $p_i \leq u$ the corresponding null hypothesis $H_{0,i}$ is rejected i.e. i-th element is considered an outlier. Threshold u is assumed here to be a fixed, predetermined point. We will write $NR(u)$ for NR to underline the dependence on u. Let False Omission Rate (FOR) be defined as

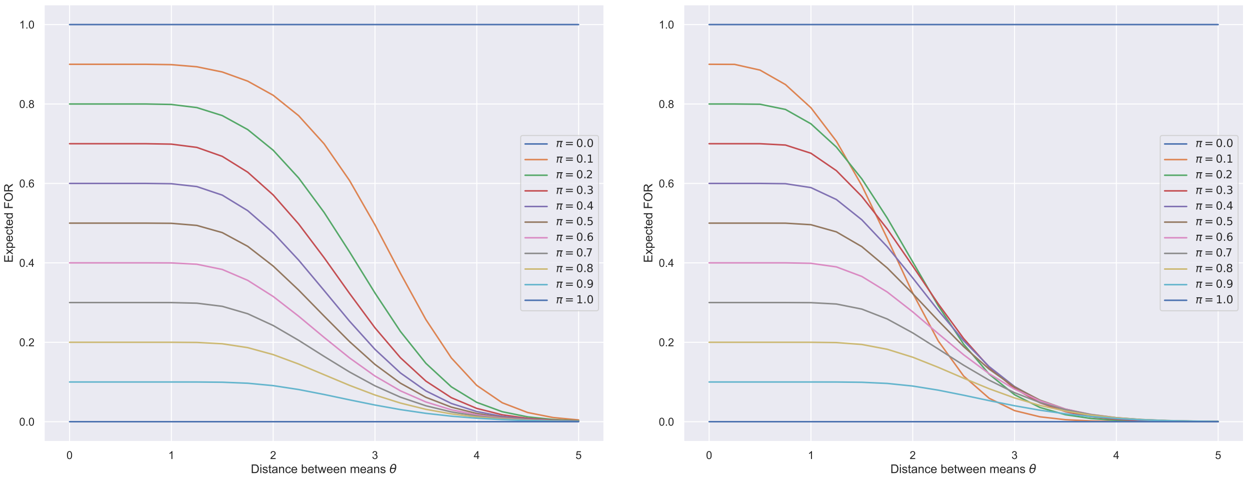
$$\text{FOR} = \mathbb{E}\left(\frac{Z}{NR(u)}\, \mathbb{I}\{NR(u) > 0\}\right), \tag{5.2}$$

Our aim is to construct a decision rule which approximately controls FOR i.e. such that for any given $\alpha \in (0,1)$ the inequality $\text{FOR} \leq \alpha$ is satisfied.

In the traditional setting one aims at controlling False Discovery Rate (FDR) at the level α, where FDR is defined as

$$\text{FDR} = \mathbb{E}\left(\frac{V}{R}\, \mathbb{I}\{R > 0\}\right). \tag{5.3}$$

We note that although it might appear at the first sight that controlling FOR defined in (5.2) is analogous to controlling FDR, this is not the case as the roles of inliers and outliers are not exchangeable. The difference is due to differences in distributions of p-values for false signals and false non-signals. Namely, we assume that the distribution of inliers' score $\widehat{s}$ is known, and it follows that for a threshold u, the distribution of the p-value corresponding to false signal, i.e. inlier smaller than u is given by the uniform distribution on $[0,u]$, whereas for the false non-signal it pertains to unknown distribution F_1 and equals $(F_1(s) - F_1(u))/(1 - F_1(u))$ for $s \in [u,1]$. As F_1 is unknown, in contrast to known (i.e. uniform) distribution of p-values for the



(a) (b)

Figure 5.1: Values of FOR when FDR is controlled against the mean distance θ between inliers and outliers (see text) (a) FDR $\leq \pi\alpha$ (Benjamini-Hochberg procedure) (b) FDR $\leq \alpha$ (Benjamini-Hochberg procedure with Storey's correction (Storey 2002)), $\alpha = 0.05$.

inliers, the problem of control of FOR is considerably harder than the control of FDR. We would like the distribution of F_1 to be concentrated close to 0, but this may vary depending on the quality of the score function in general and its performance on the studied dataset in particular (see Figure 5.5 in the results section).

We also note that similarly to testing (where decrease of level of significance leads to smaller values of power), when FDR is controlled at the level α, FOR is uncontrolled and can attain any level less than proportion of outliers $1 - \pi$.

Example 5.1. *Assume that distribution of the score statistic $\widehat{s}$ for inliers is given by the standard normal distribution $N(0,1)$ and outliers by $N(\theta,1)$, where $\theta > 0$. We reject the null for large values of s. Then straightforward calculation show that the distribution function of p-value for an outlier is given by $F_1(s) = 1 - \Phi\left(-\Phi^{-1}(s) - \theta\right) = \Phi\left(\Phi^{-1}(s) + \theta\right)$, where Φ is CDF of $N(0,1)$. Using the formula (5.6) below, the values of FOR at threshold u^*_{FDR} corresponding to Benjamini-Hochberg procedure (Benjamini and Hochberg 1995) or its modified version with Storey's correction (Storey 2002) can be calculated, and are shown in Figure 5.1. The figure shows that for small θ (when the inliers and outliers become less separated and threshold u^*_{FDR} becomes smaller), the value of FOR gets larger and approaches proportion $1 - \pi$ of inliers in the mixture.*

We also introduce Bayesian False Omission Rate (BFOR)

$$\mathrm{BFOR} = \frac{\mathbb{E}\, Z}{\mathbb{E}(NR(u))}, \tag{5.4}$$

following the analogous treatment of False Discovery Rates (see e.g. Genovese and Wasserman (2002) and Efron (2010), Section 2.2). Efron (2010) argues that from Bayesian view point, control of Bayesian False Discovery Rate – the quantity defined analogously to Eq. (5.4), but for false discoveries – is of main interest.

5.2.2 Control of FOR: theoretical results

We prove in Theorem 5.2 below that introduced quantities are approximately equal for large dataset sizes, namely

$$\text{FOR} \approx \text{BFOR}.$$

Let $G(t) = \pi U(t) + (1-\pi)F_1(t)$ be a mixture distribution of p-values. We have

Theorem 5.2. *Assume that considered decision rule rejects all null hypotheses with corresponding p-values smaller or equal u such that* $0 < G(u) < 1$. *Then*

$$FOR = BFOR \times (1-(1-G(u))^n) < BFOR.$$

Proof. We will use shorthand $p \in O$ ("p" standing for p-value and "O" standing for "Outliers") meaning that p-value corresponds to an outlier. The proof follows by noting that $P(p \in O | p > u)$ equals

$$\frac{P(p > u | p \in O) P(p \in O)}{P(p > u)} = \frac{(1-F_1(u))(1-\pi)}{1-G(u)} = \text{BFOR}. \tag{5.5}$$

Denote $\text{BFOR} = \gamma(u)$. Thus, given $NR(u)$,

$$Z | NR(u) \sim \text{Bin}(NR(u), \gamma(u)),$$

For $NR \neq 0$, using the formula for the expected value of the binomial, we have that

$$\mathbb{E}\left(\frac{Z}{NR(u)} \middle| NR(u) \right) = \frac{\mathbb{E}(Z|NR(u))}{NR(u)} = \frac{NR(u)\gamma(u)}{NR(u)} = \gamma(u) = \text{BFOR}$$

and thus

$$\begin{aligned} \text{FOR} &= \sum_{i>0} \mathbb{E}\left(\frac{Z}{NR(u)} \middle| NR(u) = i \right) P(NR(u) = i) \\ &= \sum_{i>0} \gamma(u) P(NR(u) = i) = \gamma(u) P(NR(u) \neq 0), \end{aligned}$$

which implies Theorem 5.2 as $NR(u) \sim \text{Bin}(n, 1-G(u))$.

□

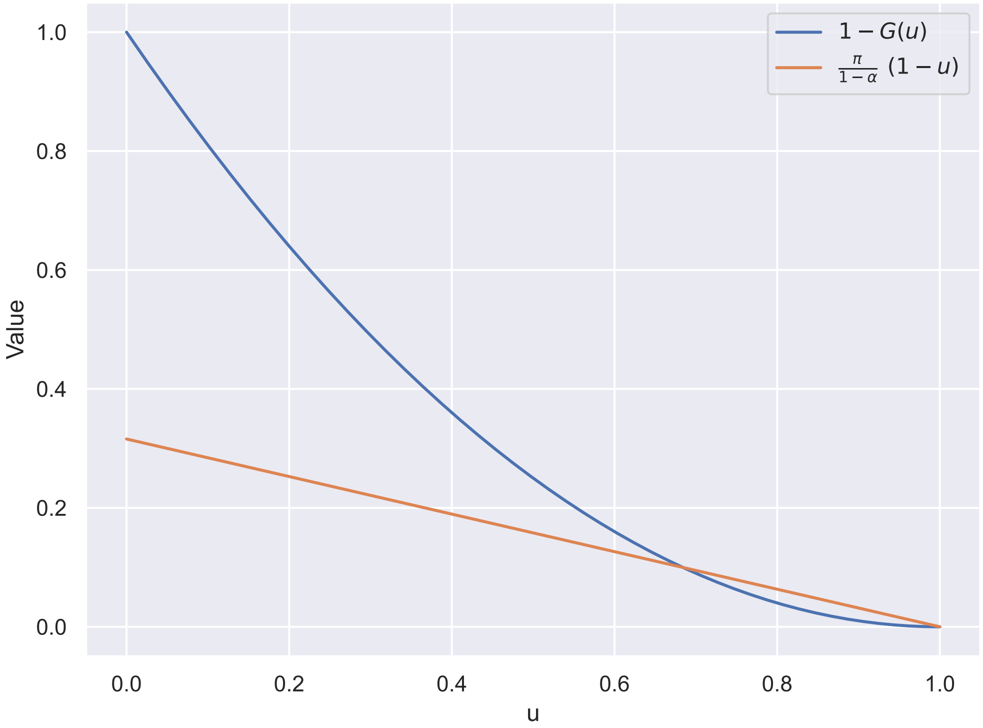


Figure 5.2: Illustration of Equation (5.8). Convex curve $1-G(u)$ starts at 0 above the value of the line $\pi/(1-\alpha)\times(1-u)$ and intersects it at a point u^*.

We note that it follows from the proof that both $Z \sim \text{Bin}\,(n,(1-\pi)(1-F_1(u))$ and $NR(u) \sim \text{Bin}(n, 1-G(u))$ are binomially distributed and thus we have

$$\frac{\mathbb{E}(Z)}{\mathbb{E}(NR(u))} = \frac{(1-\pi)(1-F_1(u))}{1-G(u)} = \frac{1-G(u)-\pi(1-u)}{1-G(u)} = P(p \in O|p > u). \qquad (5.6)$$

Note that we assume in Theorem 5.2 that threshold u does not depend on data. We conjecture that in a general case, when threshold will be data-dependent, FOR and BFOR are also approximately equivalent.

Replacing FOR in the condition FOR $= \alpha$ by its approximation BFOR and using (5.6) one obtains the following equality:

$$\frac{1-G(u)-\pi(1-u)}{1-G(u)} = \alpha, \qquad (5.7)$$

or equivalently

$$1-G(u) = \frac{\pi}{1-\alpha} \times (1-u). \qquad (5.8)$$

Theorem 5.3. *Solution $u^* \in (0,1)$ of (5.8) exists and is unique provided that (i) $G(\cdot)$ is strictly concave and (ii) $G'(1) \geq \pi/(1-\alpha)$.*

Proof. Indeed, the condition (ii) is equivalent to the condition that the derivative of $1-G(u)$ at 1 is not larger than the derivative of the line $(\pi/1-\alpha)\times(1-u)$ at 1. As $1-G(u)$ is strictly convex it is enough to check that $1-G(0)=1\geq\pi/(1-\alpha)$. But this follows from (ii) since $1>G'(1)$ as density $g(s)=G'(s)$ is strictly decreasing in view of strict concavity and $\int_0^1 g(s)\,ds=1$. Uniqueness of the solution is due to the strict concavity of G. □

Theoretical solution of (5.8) is shown in Figure 5.2. Thus we know that the truncation level u^* for such that $\text{BFOR}=\mathbb{E}(Z)/\mathbb{E}(NR(u^*))=\alpha$ exists under above conditions. Note that the assumption that $G(\cdot)$ is strictly concave (or, equivalently, that $g(\cdot)$ is strictly decreasing) is natural in the considered context. Namely, it implies in the view of (5.1) that density f_1 of p-value distribution F_1 for outliers is strictly decreasing, and, consequently, it is more likely to obtain smaller p-values for outliers than larger ones.

5.2.3 FOR control: empirical rule

Now we consider solution to the empirical counterpart of (5.8). Note that due to (5.6) BFOR is easily estimated, and we obtain the following rule: for a given $\alpha\in(0,1)$ find p-value $p_{(i^*)}$ such that

$$p_{(i^*)}=\begin{cases}\min_{p_{(i)}} B & \text{if } B\neq\emptyset;\\ 1 & \text{otherwise,}\end{cases} \tag{5.9}$$
$$\text{where } B=\left\{p_{(i)}:p_{(i)}\leq 1-\left(1-\frac{i}{n}\right)\frac{1-\alpha}{\pi}\right\}$$

and "accept" (treat as inliers) all p-values strictly larger than $p_{(i^*)}$, where $p_{(1)}\leq p_{(2)}\leq\ldots\leq p_{(n)}$. This follows by plugging in empirical distribution $G_n(t)=\#\{i:p_i\leq t\}/n$ of all p-values for G in (5.8) and noting that $G_n\left(p_{(i)}\right)=\frac{i}{n}$. Note also that the threshold in (5.9) equals 1 for $i=n$ and is approximately equal to $1-(1-\alpha)/\pi\leq 0$ for $i=1$, assuming $\alpha\leq 1-\pi$. Thus i^* is an index of the ordered p-value corresponding to the first moment when ordered p-values down-cross (cross from above to below) the line $1-(1-u)\times(1-\alpha)/\pi$. As a side node, $\alpha\leq 1-\pi$ assumption can be also justified as follows: there is a portion of $1-\pi$ of outliers in the dataset, so any threshold $alpha$ bigger than this translates to an ill-defined threshold, never rejecting any samples. Note that the rule bears significant resemblance to the Benjamini-Hochberg procedure for controlling FDR:

$$p_{(i^*)}=\begin{cases}\max_{p_{(i)}} R & \text{if } R\neq\emptyset;\\ 0 & \text{otherwise,}\end{cases} \tag{5.10}$$
$$\text{where } R=\left\{p_{(i)}:p_{(i)}\leq\frac{i}{n}\alpha\right\}$$

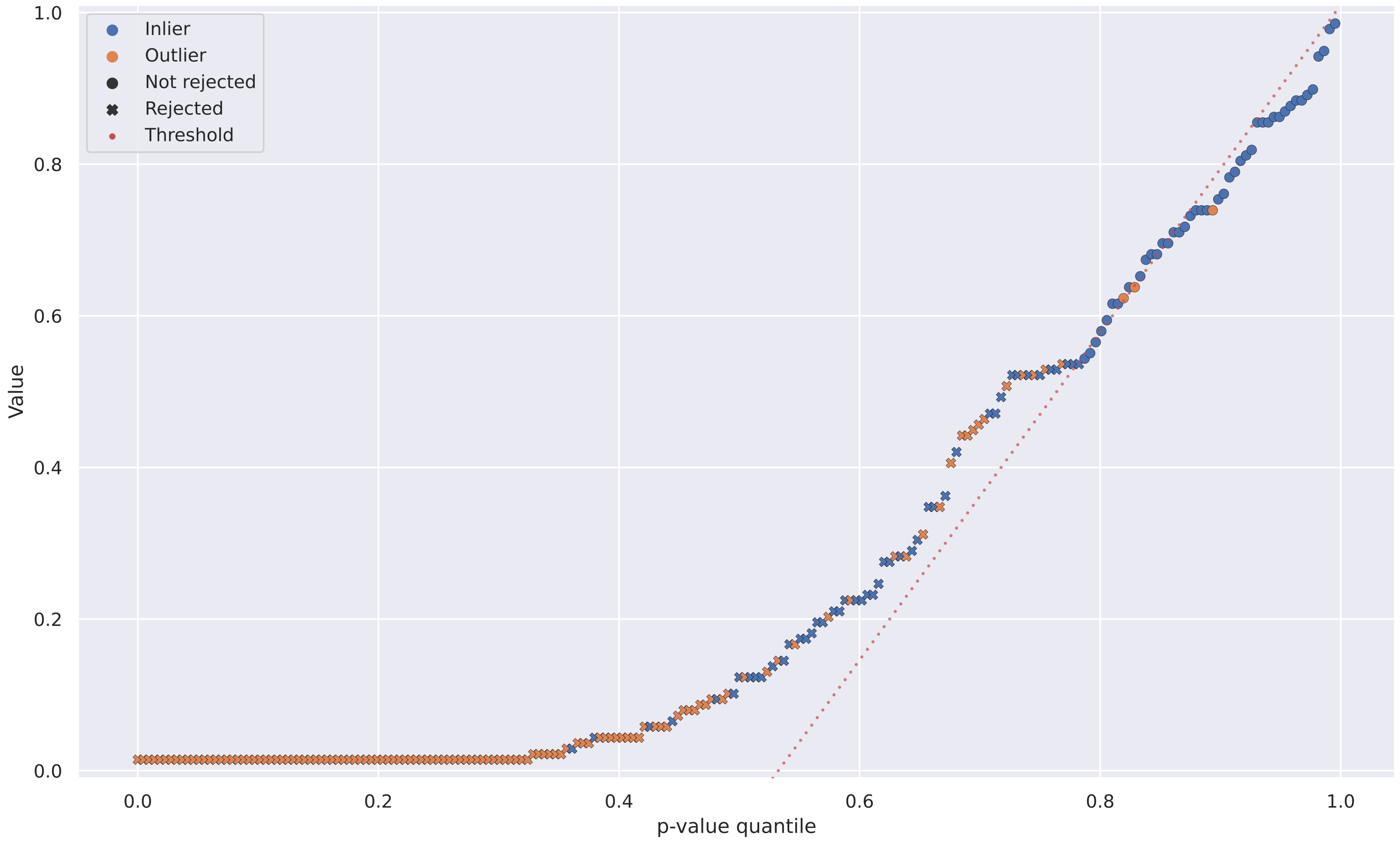


Figure 5.3: Illustration of rule (5.9). The index of the smallest p-value which down-crosses the line $1-(1-\alpha)/\pi \times (1-u)$ corresponds to the threshold in (5.9).

FOR control empirical rule (5.9) is analogous (in a symmetric way) to Benjamini-Hochberg threshold construction: starting from the largest p-values (as those are of interest when controlling not-rejected examples; this is an mirror image of Benjamini-Hochberg procedure starting from the smallest p-values) we look for the last (i.e. the smallest) index where FOR is still controlled, and use it as a threshold separating inliers from outliers.

5.2.4 Construction of p-values

We now discuss the framework in which p-values appearing in (5.9) are defined (*Multisplit* procedure). Note that as we do not know CDF of score statistic $\widehat{s}$ for inliers, we can not compute p-value directly as $1 - F(X)$ and F needs to be estimated. We thus consider a set $\mathscr{D} = \{X_1, \ldots, X_{2n}\}$ of size $2n$ consisting of inliers which will be split into training $\mathscr{D}^{\text{train}}$ and calibration $\mathscr{D}^{\text{cal}}$ sets consisting of n observations each. Moreover, let $\widehat{s}$ will be a real-valued score statistic constructed to distinguish inliers from outliers. We adopt the convention that large values of $\widehat{s}$ indicate a possible outlier. We consider the empirical distribution of $\widehat{s}(X_i)$ for $X_i \in \mathscr{D}^{\text{cal}}$ as approximation of F and define p-value $\widehat{p} = \widehat{p}(X)$ of X as

$$\widehat{p} = \frac{\#\left\{X_i \in \mathscr{D}^{\text{cal}} : \widehat{s}(X_i) \geq \widehat{s}(X)\right\} + 1}{n+1}.^{1} \tag{5.11}$$

Consider the set $S_1, \ldots, S_{n+1}$ consisting of observations $\widehat{s}(X_i)$ for $X_i \in \mathscr{D}^{\text{cal}}$ augmented by $S = \widehat{s}(X)$. When S corresponds to an inlier, observations $S_1, \ldots, S_{n+1}$ are equi-distributed and it follows that for continuous $\widehat{s}(X)$, $\widehat{p}(X)$ is uniformly distributed on $\{1/(n+1), \ldots, n/(n+1), 1\}$ given $\mathscr{D}^{\text{cal}}$, and thus $P(\widehat{p}(X) \leq t | \mathscr{D}^{\text{cal}}) \leq t$ (see e.g. Bates et al. (2023)) – this means that distribution of $p(X)$ is super-uniform.

As the definition above depends on the (random) training-calibration split, we initially considered several versions of $\widehat{p}$:

- p_{single}: one-split version defined above,
- p_{med}: median of $p_1, \ldots, p_k$ when $p_1, \ldots, p_k$ are p-values based on k random splits,
- $p_{2med} = 2 \times p_{med}$.

Definitions of p_{med} and p_{2med} are based on analogous proposals in variable selection and their purpose is to decrease variability incurred due to the random split; the phenomenon named p-value lottery (see Meinshausen, Meier, and Bühlmann (2008)). Its occurrence is confirmed by Figure 5.4 which shows substantial variability of p-values depending on a random split for A^3 classifier. The distribution of $p_{single}(X)$ is super-uniform and the same is also true for p_{2med},

[1] Note that adding 1 to both numerator and denominator results in support of $\widehat{p}$ being $\left\{\frac{1}{n+1}, \frac{2}{n+1}, \ldots, \frac{n}{n+1}, 1\right\}$.

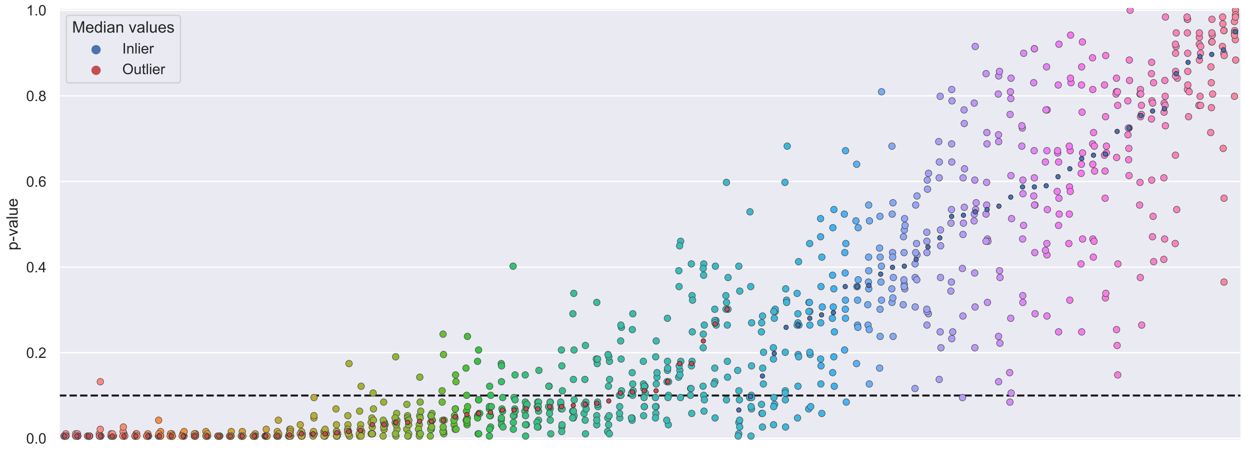


Figure 5.4: p-value lottery for A^3 classifier on 100 first examples from *Tic-tac-toe* dataset; examples are sorted according to their class and median p-value.

with the proof being analogous to that of Theorem 11.1 in Bühlmann and Geer (2011). Despite this theoretical result, Figure 5.4 shows why p_{single} should not be used directly. p-values for 10 random training-calibration splits are very unstable – median range width is 0.31, and maximal difference between minimal and maximal p-value for one of the examples exceeded 0.83. This proves that FOR control based a single split would not be reliable. p-value p_{med}, which might not be super-uniform, is also considered, as in practice p_{2med} is too conservative and thus inflates FDR in consequence. As our experiments confirm this, we focus on p_{med} only in the following.

5.3 Experimental setting

We tested the proposed FOR control procedure as described in Section 5.2.3. We consider four different score functions to obtain the outlier scores:

- Isolation Forest (Liu, Ting, and Zhou 2008) (abbreviated to *IForest*),
- Activation Anomaly Analysis A^3 (Sperl, Schulze, and Böttinger 2021) (neural network based model),
- Mahalanobis distance (Liu, Parelius, and Singh 1999) based score (abbreviated as *Mahalanobis*),
- Empirical Cumulative distribution based Outlier Detection *ECOD* (Li, Zhao, et al. 2022), as well as its variant applying ECOD to PCA-transformed data (abbreviated as *ECOD+PCA*).

For each score function, the p-values are obtained from scores using *Multisplit* procedure (Section 5.2.4), and control procedures (e.g. FOR control procedure) were applied; number of random

Table 5.1: Dataset summary

Dataset	Samples	Features	Inlier rate π
Abalone	4177	8	0.34
Arrhythmia	452	279	0.54
Banknote-auth	1372	4	0.56
Breast-w	699	9	0.66
Dermatology	366	34	0.31
Diabetes	768	8	0.65
Fertility	100	9	0.88
Gas-drift	13910	128	0.51
Glass	214	9	0.68
Haberman	306	3	0.74
Heart-statlog	270	13	0.56
Ionosphere	351	34	0.64
Isolet	7797	617	0.27
Jm1	10885	21	0.81
Kc1	2109	21	0.85
Madelon	2600	500	0.50
Musk	6598	166	0.85
Optdigits	5620	64	0.20
Pendigits	10992	16	0.20
Satimage	6430	36	0.24
Segment	2310	19	0.29
Seismic-bumps	210	7	0.33
Semeion	1593	256	0.20
Sonar	208	60	0.47
Spambase	4601	57	0.61
Tic-tac-toe	958	9	0.65
Vehicle	846	18	0.26
Waveform-5000	5000	40	0.34
Wdbc	569	30	0.63
Yeast	1484	8	0.16

splits in Multisplit procedure was set as $k = 10$. Each experiment used 60% of the inliers for training+calibration, and the remaining inliers and all of the outliers as the test set. For each test case, we repeated the entire process (starting from the training+calibration / test split) 20 times. For the control level we used $\alpha = 0.1$, which is a common value considered in literature. Code implementing the FOR control procedure, all tested methods and experiments is available publicly on GitHub[2].

Tests were conducted on 30 datasets constructed from real-world classification data. One of the classes (or several relatively similar ones) was selected as the inlier class, while other classes were considered as outliers. Basic summary of the datasets is presented in Table 5.1. For details on dataset construction, as well as their visualizations, we refer to the GitHub repository[3].

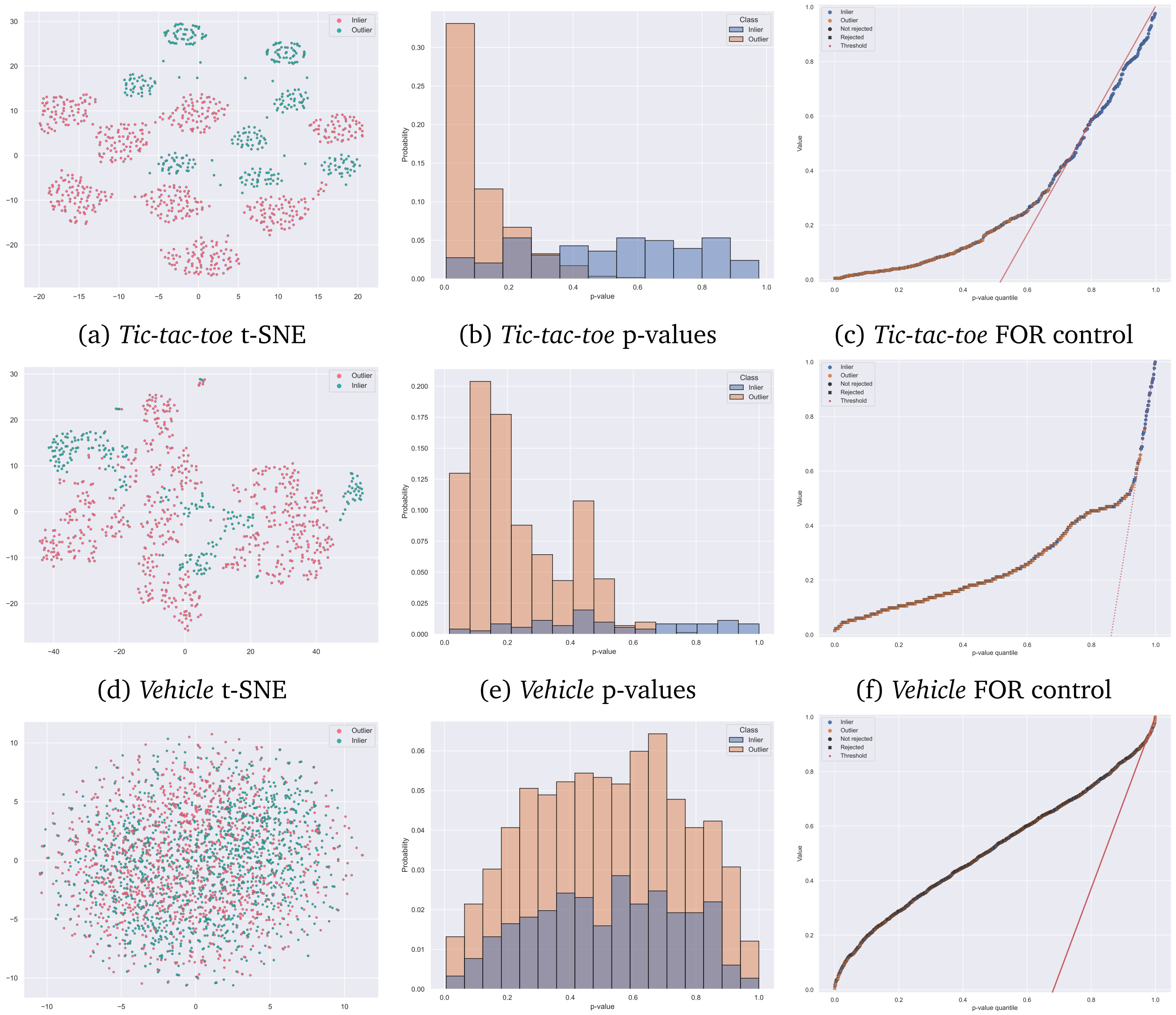
5.4 Results

Table 5.2 aggregates FOR mean values (and their standard errors) for the proposed FOR control procedure. FOR is controlled by at least one method on 20 datasets, but there are only 3 datasets where the same holds true for all methods at once. Mean FOR value was below 2α for at least one-classifier in 29 out of 30 cases (except *Yeast* dataset). Even though A^3 controlled FOR on the largest number of datasets, we will concentrate on IForest due to the higher consistency of its results. IForest managed to control FOR $\leq \alpha$ in 19 cases, FOR $\leq 2\alpha$ in 7 additional ones, and failed to keep FOR below 2α on the remaining 9 datasets.

Figure 5.5 illustrates 2-dimensional t-SNE representation of the data (1st column), distributions of the obtained p-values (2nd column) and FOR control procedure visualization (3rd column) for selected three datasets. *Tic-tac-toe* is an example of an easy dataset: we can see that

[2] `https://github.com/wawrzenczyka/FOR-CTL`

[3] `https://github.com/wawrzenczyka/FOR-CTL-datasets`



(a) *Tic-tac-toe* t-SNE (b) *Tic-tac-toe* p-values (c) *Tic-tac-toe* FOR control

(d) *Vehicle* t-SNE (e) *Vehicle* p-values (f) *Vehicle* FOR control

(g) *Madelon* t-SNE (h) *Madelon* p-values (i) *Madelon* FOR control

Figure 5.5: FOR control for datasets with varying difficulty, based on Isolation Forest scores. Red (green) dots in the first column correspond to inliers (outliers). In the second column, the orange (outlier) distribution is contrasted with the blue (inlier) distribution. The third column shows the FOR control procedure visualization, where the red line corresponds to the threshold, and each point is a p-value of an example, as depicted in Figure 5.3.

Table 5.2: FOR values and standard errors under FOR control on tested datasets, for level $\alpha = 0.1$. Magenta "✓" denotes FOR $\leq \alpha$; black "✓" denote weaker FOR $\leq 2\alpha$.

Dataset	**IForest**	A^3	**Mahalanobis**	**ECOD**	**ECOD + PCA**
Musk	0.000 ± 0.000✓✓	0.137 ± 0.005✓	0.415 ± 0.040	0.000 ± 0.000✓✓	0.135 ± 0.011✓
Seismic-bumps	0.061 ± 0.018✓✓	0.093 ± 0.028✓✓	0.056 ± 0.018✓✓	0.048 ± 0.016✓✓	0.099 ± 0.025✓✓
Ionosphere	0.067 ± 0.014✓✓	0.107 ± 0.016✓	0.082 ± 0.011✓✓	0.081 ± 0.016✓✓	0.140 ± 0.009✓
Tic-tac-toe	0.075 ± 0.008✓✓	0.061 ± 0.004✓✓	0.140 ± 0.023✓	0.117 ± 0.012✓	0.272 ± 0.030
Breast-w	0.076 ± 0.005✓✓	0.083 ± 0.003✓✓	0.086 ± 0.004✓✓	0.057 ± 0.005✓✓	0.095 ± 0.004✓✓
Isolet	0.078 ± 0.006✓✓	0.047 ± 0.014✓✓	0.282 ± 0.006	0.090 ± 0.009✓✓	0.363 ± 0.004
Dermatology	0.078 ± 0.013✓✓	0.037 ± 0.006✓✓	0.049 ± 0.014✓✓	0.052 ± 0.008✓✓	0.206 ± 0.022
Semeion	0.078 ± 0.014✓✓	0.074 ± 0.012✓✓	0.049 ± 0.012✓✓	0.118 ± 0.016✓	0.000 ± 0.000✓✓
Banknote-auth	0.079 ± 0.009✓✓	0.086 ± 0.029✓✓	0.078 ± 0.003✓✓	0.069 ± 0.012✓✓	0.085 ± 0.006✓✓
Pendigits	0.091 ± 0.005✓✓	0.091 ± 0.004✓✓	0.108 ± 0.006✓	0.100 ± 0.012✓✓	0.099 ± 0.010✓✓
Satimage	0.094 ± 0.004✓✓	0.000 ± 0.000✓✓	0.084 ± 0.005✓✓	0.129 ± 0.010✓	0.105 ± 0.007✓
Segment	0.094 ± 0.007✓✓	0.317 ± 0.082	0.109 ± 0.009✓	0.085 ± 0.005✓✓	0.151 ± 0.017✓
Kc1	0.094 ± 0.008✓✓	0.091 ± 0.008✓✓	0.089 ± 0.009✓✓	0.170 ± 0.022✓	0.129 ± 0.007✓
Wdbc	0.097 ± 0.009✓✓	0.088 ± 0.010✓✓	0.100 ± 0.006✓✓	0.097 ± 0.010✓✓	0.137 ± 0.007✓
Optdigits	0.101 ± 0.010✓	0.074 ± 0.011✓✓	0.083 ± 0.008✓✓	0.160 ± 0.036✓	0.188 ± 0.012✓
Gas-drift	0.114 ± 0.010✓	0.000 ± 0.000✓✓	0.146 ± 0.011✓	0.088 ± 0.014✓✓	0.143 ± 0.013✓
Spambase	0.122 ± 0.021✓	0.139 ± 0.008✓	0.140 ± 0.017✓	0.217 ± 0.056	0.173 ± 0.025✓
Vehicle	0.127 ± 0.024✓	0.000 ± 0.000✓✓	0.073 ± 0.009✓✓	0.160 ± 0.045✓	0.162 ± 0.014✓
Glass	0.147 ± 0.030✓	0.163 ± 0.025✓	0.121 ± 0.019✓	0.186 ± 0.039✓	0.156 ± 0.018✓
Heart-statlog	0.153 ± 0.020✓	0.277 ± 0.035	0.170 ± 0.021✓	0.140 ± 0.026✓	0.246 ± 0.050
Diabetes	0.179 ± 0.017✓	0.151 ± 0.021✓	0.240 ± 0.030	0.271 ± 0.079	0.133 ± 0.023✓
Waveform-5000	0.204 ± 0.020	0.264 ± 0.076	0.161 ± 0.041✓	0.142 ± 0.024✓	0.355 ± 0.046
Abalone	0.206 ± 0.008	0.093 ± 0.014✓✓	0.170 ± 0.015✓	0.302 ± 0.030	0.167 ± 0.017✓
Arrhythmia	0.214 ± 0.037	0.187 ± 0.076✓	0.197 ± 0.030✓	0.300 ± 0.058	0.292 ± 0.017
Fertility	0.219 ± 0.024	0.121 ± 0.016✓	0.218 ± 0.024	0.229 ± 0.036	0.223 ± 0.022
Yeast	0.228 ± 0.069	0.299 ± 0.088	0.226 ± 0.078	0.301 ± 0.084	0.251 ± 0.092
Jm1	0.237 ± 0.008	0.000 ± 0.000✓✓	0.189 ± 0.022✓	0.312 ± 0.027	0.242 ± 0.020
Haberman	0.293 ± 0.035	0.472 ± 0.057	0.366 ± 0.058	0.153 ± 0.033✓	0.259 ± 0.039
Sonar	0.431 ± 0.078	0.125 ± 0.047✓	0.252 ± 0.052	0.175 ± 0.083✓	0.506 ± 0.030
Madelon	0.748 ± 0.013	0.495 ± 0.010	0.722 ± 0.008	0.100 ± 0.069✓✓	0.150 ± 0.082✓

the data forms distinct, separate clusters (Fig. 5.5a) and therefore there is a clear difference in inlier and outlier p-value distributions (Fig. 5.5b), as well as nearly perfectly uniform inlier distribution – IForest captured the inlier distribution really well. In that case, FOR control procedure has no issues capturing the clean portion of inliers (Fig. 5.5c) with the occasional outlier examples allowed by the α parameter.

Vehicle dataset in the second row is of medium difficulty: inlier and outlier examples (Fig. 5.5d) are a lot more difficult to separate. Note that the outlier p-value distribution (Fig. 5.5e) is shifted right, towards higher values, and inlier distribution is not as regular as in previous case. Though this example is significantly harder, we can see that in this particular case FOR control (Fig. 5.5f) divided the examples perfectly – though multiple outlier examples are extremely close to the threshold and might be incorrectly undetected with a small variations in their p-values. That leads to FOR for this dataset in Table 5.2 being slightly higher than desired.

Madelon dataset, on the other hand, was selected as an hard problem example. T-SNE visualization (Fig. 5.5g) does not capture any visible outlier characteristics, which suggests that the relationships in the data are complex. As a consequence, p-value distributions obtained from IForest scores (Fig. 5.5h) are extremely similar between outliers and inliers; note that the inlier

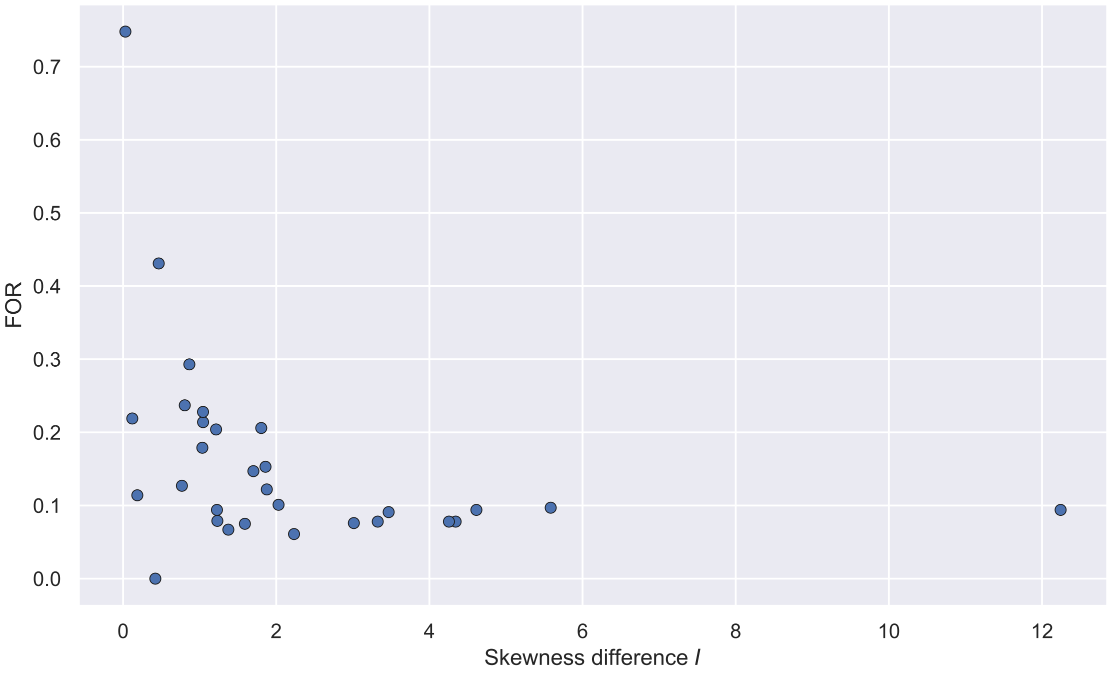


Figure 5.6: Mean FOR values versus mean skewness difference I between outlier and inlier p-value distributions; each dot on the plot corresponds to one dataset.

p-value density is slightly bell shaped and thus deviates from the uniform, moreover the outliers p-value density does not decrease. As the proposed procedure assumes those properties, their lack has a profound impact on the FOR control (Fig. 5.5i). Lower than expected (when uniformity holds) number of inlier examples with high p-values causes outliers with high p-values to take their place; this results in the dramatic omission of dominant part of outlier examples, which yield a very high FOR value, as presented in Tab. 5.2. We note that deviation from uniformity for inlier p-values may be due to the fact that for this synthetic dataset inliers are generated from multimodal distribution, with modes being the vertices of a high-dimensional hypercube (Guyon 2004).

Figure 5.5 suggests that obtaining high quality scores (and, as a result, reliable p-values) from the classifier is fundamental in order to ensure a good FOR control. That makes p-value distribution properties worth inspecting. In particular, we explored the effect of difference in skewnesses I of outlier and inlier p-value distribution, given by formula $I = Skew_{OUT} - Skew_{IN}$, on the empirical FOR value. We expect inlier distribution to be uniform (so $Skew_{IN}$ should be close to 0), whereas probability mass for the outlier distribution should be concentrated on small p-values (resulting in large positive values of $Skew_{OUT}$), which should mean that for a p-value distributions satisfying the imposed assumptions, I should be both positive and relatively large. Indeed, as illustrated in the Figure 5.6, datasets where I is large are also the ones where the proposed procedure works really well; on the other hand, when I falls below 2, FOR control

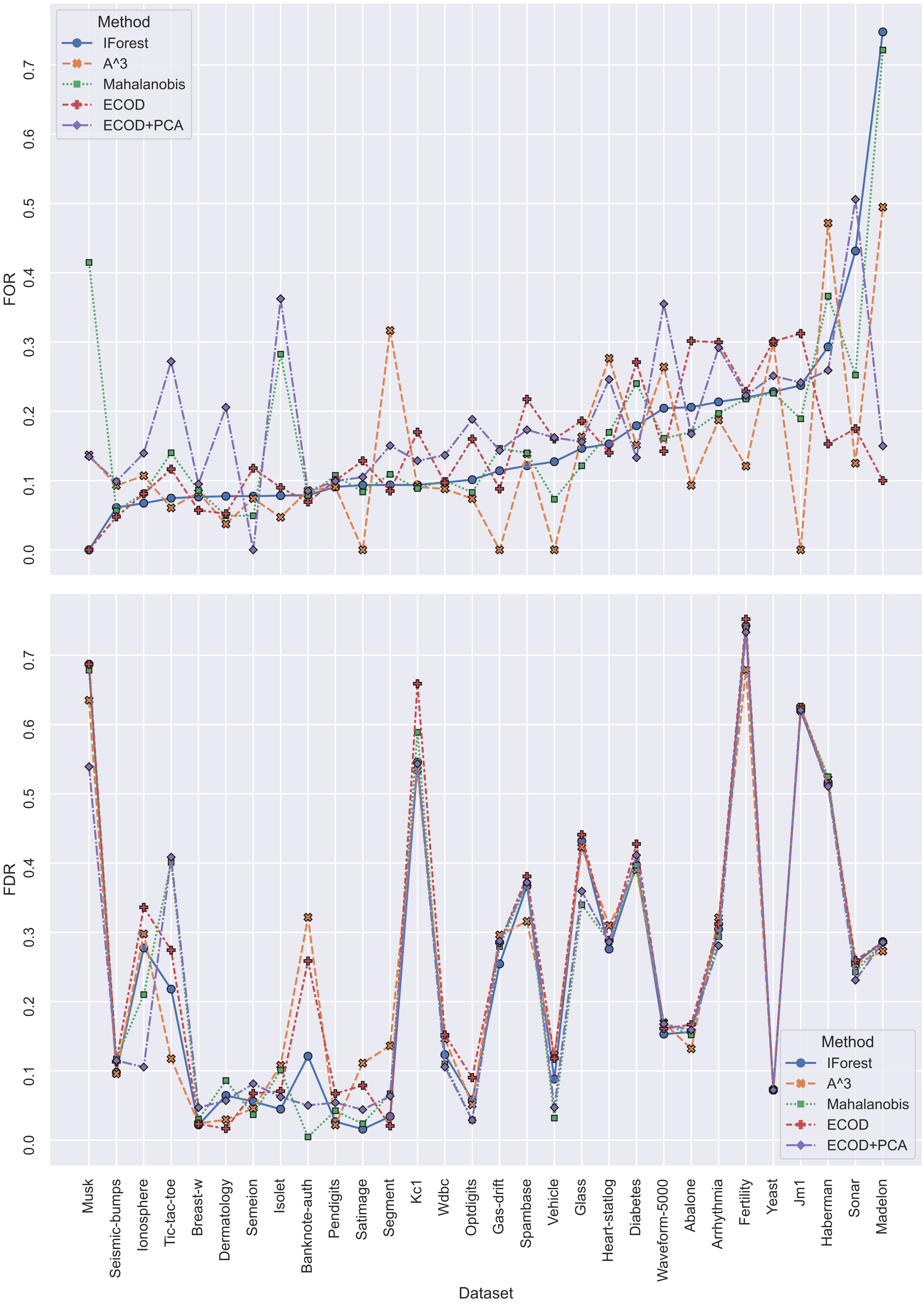


Figure 5.7: FOR values for all methods and data sets (upper panel) with the corresponding values of FDR (lower panel)

becomes unreliable. This emphasizes the dependency of FOR control on outlier score quality – we can control FOR only if outlier scores make that possible.

Figure 5.7 visualizes relationship between FOR and FDR when FOR is controlled on all sets. Observe that good control of FOR doesn't imply low FDR value (and vice versa, see Figure 5.1). Moreover, this holds irrespectively of the scoring method – even when a given method controls FOR at a given level, its FDR might remain high.

5.5 VAE-PU+FOR

Using FOR control procedure described in Section 5.2.3 we can also control FOR for the VAE-PU+OCC method. As described in the previous chapter, VAE-PU+OCC presents us with an inner one-class classification problem, and the equivalent outlier detection task. In this task, the term "False Omission" refers to negative unlabeled examples present in the synthetic positive unlabeled dataset, created in step 7 of Algorithm 2. By controlling FOR, we control the purity of the inlier set, which, translated to the VAE-PU context, means the obtained candidate PU set – by using FOR control procedure, we want to ensure that the proportion of negative unlabeled examples in the synthetic positive unlabeled dataset is not larger than α, where α is a hyperparameter. The resultant VAE-PU+FOR method is described in Algorithm 3.

```
Algorithm 3: VAE-PU+FOR training
Input: $\pi$ – class prior, $n$ – number of training items
Train VAE-PU model (encoder, decoder, target classifier, observation classifier and
  discriminator); this process is described in detail in Section 4.4;
while not converged do
    Generate pseudo-set $\widetilde{x}_{PU}$ using trained VAE;
    Train OCC classifier of choice and prepare Multisplit calibration scores on $\widetilde{x}_{PU}$;
    Use OCC classifier to calculate Multisplit p-values for U dataset;
    Fit FOR control procedure to find the threshold $p_{(i^*)}$ ensuring that FOR is controlled
      at level $\alpha$;
    Choose all examples in U with p-values smaller than $p_{(i^*)}$ as the candidate PU set;
    Update VAE-PU target classifier with the risk function $R_{emp}(g)$ and candidate PU set;
end
```

Crucially, note that the max term omission discussed in the context of Equation (4.40) is no longer valid for algorithm 3. Avoiding loss clipping by dropping the max term was an important building block of the VAE-PU+OCC, and was enabled by a proper control of the selected positive unlabeled set size, which is no longer possible when FOR is controlled instead. We sacrifice it in hopes that the purer and higher quality PU set will make up for the deterioration in the optimization process.

Table 5.3: Accuracy of VAE-PU+FOR for varying control levels α, contrasted with VAE-PU+IForest method.

	α	**0.01**	**0.05**	**0.1**	**0.2**	**VAE-PU+IForest**
Dataset	c					
CDC-Diabetes	0.02	**0.543 ± 0.013**	0.541 ± 0.013	0.536 ± 0.013	0.539 ± 0.012	0.524 ± 0.011
	0.10	0.587 ± 0.009	**0.592 ± 0.010**	0.589 ± 0.013	0.581 ± 0.008	0.571 ± 0.015
	0.30	0.667 ± 0.004	0.674 ± 0.004	**0.678 ± 0.003**	0.660 ± 0.004	0.675 ± 0.004
	0.50	0.702 ± 0.003	0.706 ± 0.002	**0.709 ± 0.002**	0.697 ± 0.003	0.706 ± 0.002
	0.70	0.715 ± 0.001	0.718 ± 0.001	**0.720 ± 0.001**	0.711 ± 0.002	0.719 ± 0.002
	0.90	0.718 ± 0.001	0.720 ± 0.001	0.722 ± 0.001	0.716 ± 0.001	**0.725 ± 0.001**
CIFAR CT	0.02	0.932 ± 0.002	0.935 ± 0.002	0.936 ± 0.001	**0.937 ± 0.002**	0.937 ± 0.001
	0.10	0.935 ± 0.002	0.937 ± 0.002	0.937 ± 0.001	0.935 ± 0.002	**0.938 ± 0.001**
	0.30	0.938 ± 0.002	0.939 ± 0.002	0.941 ± 0.001	0.939 ± 0.002	**0.942 ± 0.001**
	0.50	0.922 ± 0.004	0.937 ± 0.003	0.941 ± 0.002	0.939 ± 0.002	**0.942 ± 0.002**
	0.70	0.897 ± 0.009	0.925 ± 0.006	0.938 ± 0.005	0.939 ± 0.002	**0.943 ± 0.002**
	0.90	0.893 ± 0.008	0.917 ± 0.007	0.935 ± 0.002	0.944 ± 0.002	**0.945 ± 0.002**
CIFAR VA	0.02	0.926 ± 0.002	0.932 ± 0.002	0.932 ± 0.002	0.926 ± 0.003	**0.933 ± 0.002**
	0.10	0.905 ± 0.005	0.919 ± 0.003	**0.932 ± 0.001**	0.920 ± 0.003	0.926 ± 0.001
	0.30	0.904 ± 0.006	0.921 ± 0.005	**0.940 ± 0.002**	0.927 ± 0.005	0.934 ± 0.001
	0.50	0.917 ± 0.003	0.928 ± 0.004	0.941 ± 0.001	**0.945 ± 0.001**	0.938 ± 0.001
	0.70	0.913 ± 0.004	0.930 ± 0.004	0.937 ± 0.003	**0.947 ± 0.002**	0.940 ± 0.001
	0.90	0.913 ± 0.006	0.932 ± 0.003	0.945 ± 0.002	0.943 ± 0.003	**0.948 ± 0.001**
MNIST 3v5	0.02	0.780 ± 0.009	0.780 ± 0.010	0.781 ± 0.011	**0.781 ± 0.010**	0.778 ± 0.011
	0.10	0.891 ± 0.005	0.895 ± 0.005	0.896 ± 0.005	0.887 ± 0.005	**0.899 ± 0.005**
	0.30	0.922 ± 0.002	0.924 ± 0.002	0.926 ± 0.003	0.920 ± 0.002	**0.926 ± 0.003**
	0.50	0.938 ± 0.004	0.940 ± 0.003	**0.940 ± 0.003**	0.937 ± 0.003	0.940 ± 0.003
	0.70	0.943 ± 0.004	0.944 ± 0.004	**0.945 ± 0.004**	0.943 ± 0.004	0.943 ± 0.005
	0.90	0.953 ± 0.003	0.955 ± 0.002	**0.956 ± 0.002**	0.953 ± 0.003	0.955 ± 0.002
MNIST OvE	0.02	0.806 ± 0.011	0.810 ± 0.011	**0.815 ± 0.011**	0.803 ± 0.011	0.812 ± 0.012
	0.10	0.855 ± 0.010	0.855 ± 0.008	0.859 ± 0.009	0.850 ± 0.010	**0.862 ± 0.010**
	0.30	0.898 ± 0.003	0.893 ± 0.006	0.904 ± 0.003	0.897 ± 0.005	**0.908 ± 0.003**
	0.50	0.921 ± 0.003	0.921 ± 0.003	0.921 ± 0.003	0.916 ± 0.003	**0.926 ± 0.002**
	0.70	0.938 ± 0.003	0.940 ± 0.003	0.941 ± 0.002	0.936 ± 0.003	**0.944 ± 0.003**
	0.90	0.959 ± 0.003	0.961 ± 0.002	0.964 ± 0.001	0.955 ± 0.004	**0.965 ± 0.001**
STL VA	0.02	0.846 ± 0.004	0.846 ± 0.004	0.847 ± 0.006	0.843 ± 0.004	**0.849 ± 0.006**
	0.10	0.855 ± 0.004	0.866 ± 0.003	0.876 ± 0.003	0.870 ± 0.004	**0.879 ± 0.003**
	0.30	0.856 ± 0.004	0.876 ± 0.004	0.889 ± 0.003	0.886 ± 0.004	**0.891 ± 0.004**
	0.50	0.867 ± 0.002	0.880 ± 0.002	**0.900 ± 0.002**	0.898 ± 0.003	0.897 ± 0.003
	0.70	0.850 ± 0.004	0.881 ± 0.003	0.900 ± 0.003	0.899 ± 0.002	**0.903 ± 0.003**
	0.90	0.879 ± 0.005	0.893 ± 0.003	0.904 ± 0.003	0.899 ± 0.004	**0.912 ± 0.003**

Table 5.4: F1 score of VAE-PU+FOR for varying control levels α, contrasted with VAE-PU+IForest method.

	α	**0.01**	**0.05**	**0.1**	**0.2**	**VAE-PU+IForest**
Dataset	c					
CDC-Diabetes	0.02	0.615 ± 0.006	0.637 ± 0.004	0.657 ± 0.002	0.593 ± 0.009	**0.666 ± 0.003**
	0.10	0.628 ± 0.006	0.646 ± 0.005	0.662 ± 0.004	0.613 ± 0.007	**0.673 ± 0.003**
	0.30	0.684 ± 0.004	0.693 ± 0.004	0.704 ± 0.003	0.675 ± 0.005	**0.715 ± 0.003**
	0.50	0.716 ± 0.003	0.722 ± 0.003	0.730 ± 0.003	0.710 ± 0.004	**0.740 ± 0.002**
	0.70	0.723 ± 0.001	0.726 ± 0.001	0.735 ± 0.001	0.721 ± 0.002	**0.746 ± 0.002**
	0.90	0.727 ± 0.001	0.723 ± 0.002	0.724 ± 0.002	0.724 ± 0.001	**0.746 ± 0.002**
CIFAR CT	0.02	0.933 ± 0.002	0.935 ± 0.002	0.936 ± 0.002	**0.937 ± 0.002**	0.936 ± 0.001
	0.10	0.935 ± 0.002	0.937 ± 0.002	0.936 ± 0.002	0.934 ± 0.002	**0.937 ± 0.002**
	0.30	0.938 ± 0.002	0.938 ± 0.002	0.940 ± 0.002	0.938 ± 0.002	**0.940 ± 0.002**
	0.50	0.925 ± 0.004	0.938 ± 0.003	0.940 ± 0.002	0.937 ± 0.002	**0.941 ± 0.002**
	0.70	0.905 ± 0.008	0.927 ± 0.006	0.938 ± 0.005	0.937 ± 0.002	**0.942 ± 0.002**
	0.90	0.901 ± 0.007	0.921 ± 0.006	0.937 ± 0.002	0.943 ± 0.002	**0.944 ± 0.002**
CIFAR VA	0.02	0.909 ± 0.003	**0.915 ± 0.002**	0.913 ± 0.003	0.905 ± 0.003	0.913 ± 0.003
	0.10	0.889 ± 0.005	0.903 ± 0.003	**0.916 ± 0.002**	0.897 ± 0.004	0.902 ± 0.002
	0.30	0.890 ± 0.005	0.907 ± 0.005	**0.926 ± 0.002**	0.906 ± 0.006	0.912 ± 0.002
	0.50	0.904 ± 0.003	0.914 ± 0.004	0.928 ± 0.001	**0.931 ± 0.001**	0.917 ± 0.001
	0.70	0.900 ± 0.004	0.917 ± 0.005	0.924 ± 0.003	**0.933 ± 0.002**	0.921 ± 0.002
	0.90	0.900 ± 0.007	0.919 ± 0.003	**0.933 ± 0.002**	0.931 ± 0.003	0.932 ± 0.001
MNIST 3v5	0.02	0.803 ± 0.008	0.798 ± 0.009	0.802 ± 0.008	0.799 ± 0.008	**0.808 ± 0.008**
	0.10	0.898 ± 0.004	0.901 ± 0.004	0.903 ± 0.005	0.895 ± 0.004	**0.906 ± 0.005**
	0.30	0.927 ± 0.002	0.928 ± 0.002	0.930 ± 0.002	0.924 ± 0.001	**0.931 ± 0.003**
	0.50	0.942 ± 0.003	0.943 ± 0.003	**0.944 ± 0.003**	0.940 ± 0.003	0.943 ± 0.003
	0.70	0.947 ± 0.004	**0.948 ± 0.004**	0.948 ± 0.004	0.946 ± 0.003	0.946 ± 0.004
	0.90	0.956 ± 0.002	0.957 ± 0.002	**0.958 ± 0.002**	0.955 ± 0.002	0.957 ± 0.002
MNIST OvE	0.02	0.822 ± 0.011	0.822 ± 0.009	0.825 ± 0.008	**0.825 ± 0.008**	0.818 ± 0.012
	0.10	**0.866 ± 0.008**	0.861 ± 0.006	0.861 ± 0.009	0.863 ± 0.008	0.864 ± 0.010
	0.30	0.902 ± 0.003	0.894 ± 0.007	0.904 ± 0.003	0.903 ± 0.004	**0.908 ± 0.003**
	0.50	0.924 ± 0.003	0.922 ± 0.003	0.920 ± 0.003	0.921 ± 0.003	**0.926 ± 0.002**
	0.70	0.940 ± 0.003	0.942 ± 0.003	0.941 ± 0.003	0.939 ± 0.003	**0.944 ± 0.003**
	0.90	0.960 ± 0.003	0.962 ± 0.002	0.964 ± 0.001	0.957 ± 0.004	**0.965 ± 0.001**
STL VA	0.02	0.784 ± 0.006	0.785 ± 0.005	0.790 ± 0.007	0.781 ± 0.006	**0.798 ± 0.008**
	0.10	0.835 ± 0.003	0.842 ± 0.002	**0.844 ± 0.004**	0.826 ± 0.005	0.839 ± 0.004
	0.30	0.839 ± 0.004	0.856 ± 0.004	**0.864 ± 0.003**	0.848 ± 0.006	0.854 ± 0.005
	0.50	0.850 ± 0.002	0.860 ± 0.002	**0.876 ± 0.002**	0.867 ± 0.004	0.862 ± 0.004
	0.70	0.837 ± 0.003	0.863 ± 0.003	**0.879 ± 0.003**	0.875 ± 0.003	0.871 ± 0.003
	0.90	0.862 ± 0.004	0.874 ± 0.002	0.883 ± 0.003	0.879 ± 0.004	**0.888 ± 0.003**

Table 5.5: Ratio of examples selected by VAE-PU+FOR to the number of examples selected by VAE-PU+IForest ($|\widetilde{\chi}_{PU_{\text{FOR}}}|/|\widetilde{\chi}_{PU_{\text{IForest}}}| \approx |\widetilde{\chi}_{PU_{\text{FOR}}}|/|\chi_{PU}|$), for varying control levels α.

	α	**0.01**	**0.05**	**0.1**	**0.2**
Dataset	c				
CDC-Diabetes	0.02	0.892	0.932	0.972	1.099
	0.10	0.926	0.965	1.087	1.220
	0.30	0.969	1.026	1.076	1.228
	0.50	0.951	1.009	1.069	1.216
	0.70	0.978	1.018	1.075	1.190
	0.90	0.968	0.999	1.071	1.198
CIFAR CT	0.02	0.470	0.504	0.726	0.958
	0.10	0.326	0.649	0.976	1.105
	0.30	0.033	0.401	0.972	1.106
	0.50	0.025	0.118	0.980	1.113
	0.70	0.017	0.066	0.874	1.110
	0.90	0.015	0.110	0.311	1.069
CIFAR VA	0.02	0.215	0.236	0.338	0.900
	0.10	0.206	0.255	0.312	0.525
	0.30	0.172	0.214	0.252	0.420
	0.50	0.052	0.062	0.189	0.291
	0.70	0.048	0.070	0.142	0.263
	0.90	0.054	0.081	0.146	0.230
MNIST 3v5	0.02	0.554	0.594	0.684	0.868
	0.10	0.526	0.578	0.680	0.869
	0.30	0.514	0.557	0.638	0.848
	0.50	0.501	0.551	0.629	0.831
	0.70	0.439	0.486	0.543	0.689
	0.90	0.270	0.285	0.320	0.379
MNIST OvE	0.02	0.020	0.021	0.028	0.032
	0.10	0.009	0.009	0.012	0.015
	0.30	0.004	0.005	0.006	0.007
	0.50	0.006	0.006	0.010	0.018
	0.70	0.004	0.004	0.007	0.012
	0.90	0.009	0.012	0.014	0.016
STL VA	0.02	0.723	0.753	0.880	0.998
	0.10	0.268	0.343	0.456	0.607
	0.30	0.202	0.279	0.365	0.523
	0.50	0.221	0.278	0.341	0.458
	0.70	0.148	0.224	0.275	0.386
	0.90	0.122	0.142	0.174	0.240

We conducted a set of experiments comparing the VAE-PU+FOR method with the VAE-PU+OCC method, utilizing the best Isolation Forest classifier as the outlier detection method. The experimental setting is unchanged from the one in the previous chapter, except for a new medical dataset replacing Gas Concentrations, CDC Diabetes. Note that starting from this chapter, we also use shorthands "CIFAR CT" and "CIFAR VA" for CIFAR CarTruck and MachineAnimal datasets, respectively (and similarly, "STL VA"). For details about datasets and labeling, refer to Appendix E. We used four different settings for the control level α: 0.01, 0.05, 0.1 and 0.2, which serve here as values of a hyperparameter. The method and all experiment code is available in a public Github repository[4].

The results are summarized in Tables 5.3 and 5.4. Both tables lead to a similar conclusion: the VAE-PU+FOR method's performance is close to VAE-PU+IForest in terms of both accuracy and F1 score, with rare marginal improvements (the comparison of VAE-PU+IForest with standard VAE-PU can be found in the previous chapter; as VAE-PU+IForest generally outperforms the baseline model, we skip it in the comparison). The results are consistent across all datasets and control levels, with control level $\alpha = 0.1$ being the best – it offers the best trade-off between the size of the candidate PU set and its quality. However, the VAE-PU+IForest method usually outperforms its FOR-controlled counterpart. There are many potential causes of this performance decrease – the loss clipping might impact method's performance negatively, and as evident in the prior FOR control results, the procedure requires a good p-value estimates to work properly, which makes it very sensitive to the OCC model performance. An important observation is that the VAE-PU+FOR method is more conservative than a pure VAE-PU+IForest method, as it tends to select less candidate PU examples. This is evident in Table 5.5, presenting the ratio of selected PU examples to the number of examples chosen as positive from unlabled data set by VAE-PU+IForest (which is equal to the expected number of PU examples present in the U set). Except for the CDC-Diabetes dataset, the VAE-PU+FOR method selected a smaller number of PU candidate examples than present – sometimes even drastically less, as in the MNIST OvE case. Normally, the number of selected examples decreases with both lower control level and higher label frequency. The number of selected examples might be too low for proper PU risk estimation (especially in the edge cases, such as MNIST OvE), and FOR control procedure based on the imperfect VAE-PU generated examples might be too conservative.

5.6 Conclusions

In this chapter we propose the first (to our knowledge) empirical procedure allowing for FOR control in the outlier detection scenario, which is vital in many real-life scenarios. Our approach

[4] https://github.com/wawrzenczyka/VAE-PU-FOR

is mathematically justified and accounts for prior research on the related control algorithms, such as Benjamini-Hochberg procedure for FDR control and its empirical Bayes underpinning. It is important to note that the FOR control problem is substantially harder than its FDR counterpart, due to threshold calculation requiring outlier distribution properties. The experiments presented in the chapter prove method's capability of controlling FOR as long as good quality p-values are provided to the algorithm. That ties into the most significant limitation of the described procedure – the dependence on the outlier scores supplied by the external methods makes their imperfections transfer to the researched task. FOR control procedure is sensitive to breaking its assumptions – this is most evident when skewness difference between outlier and inlier p-value distribution is low, which results in outliers replacing a portion of missing inlier distribution, which in turn causes their uncontrolled omission.

The last part of the chapter presents a new method, VAE-PU+FOR, which combines the VAE-PU method with the FOR control procedure. However, despite the sound purity control improvement promise, in practice the method does not outperform the VAE-PU+OCC method described in the previous chapter. The results suggest that the FOR control procedure is too conservative, and the loss clipping might be detrimental to the optimization process. The method is also sensitive to the quality of the outlier scores, which makes it difficult to use in practice.

Chapter 6

Augmented PU learning

6.1 Introduction

In the previous chapters we have considered the general, no-SCAR PU learning problem, where some observations from a positive class are assigned labels, whereas the remaining observations from this class, as well as all negative observations, are unlabeled, and labeling may depend on covariates. Under such scenario the most common Machine Learning task is construction of a classification rule based *only* on predictors, which will assign a new observation to a positive or a negative class – many examples of such methods have been described in the introduction. However, there are PU learning problems where the data partial observability can be slightly loosened – in many applications we would like to perform classification on *new* PU observations for which along with predictors, *the labeling status is given*.

Such situations commonly happen. A typical example is occurrence of hypertension. People who check their blood pressure regularly and when it is abnormal report that to a doctor, are treated for hypertension. In such a case positive labels are assigned to them. However, the remaining (unlabeled) group consists of those who have abnormal blood pressure level but do not contact a doctor, and those who are healthy. Another example is reporting episodes of certain illness (i.e. migraine) using dedicated software, see e.g. Park et al. (2016). Here, some patients who experience such episodes, fail to report them and thus they can not be distinguished from patients who do not have them, whence both groups fall into unlabeled category. Note that in the considered examples labeling status of new observations is naturally known. In the first example above, in a new batch of patients, for those who fail to report hypertension, one would like to detect those who are likely to be positive – we know for a fact that each of these patients is unlabeled, as they did not report the health problems, labeled examples correspond to previous illness reports. Similarly, in the "migraine" example it is of interest to detect patients who likely have failed to report migraine episodes, in order to contact them, here also knowing whether the migraine episode was reported or not. Of course, in such a case, assignment is an issue for unlabeled observations only, as for the labeled ones we know for sure that they belong to a positive class. For the sake of distinguishing such task from the usual classification based on predictors alone, we will call this problem prediction for augmented PU observations or, in short, *augmented PU prediction*. To the best of our knowledge this is the first approach discussing this problem in the literature.

In this chapter we establish the form of Bayes selection rule for detection of positive observations among unlabeled ones and show that it is more conservative than Bayes classification rule based solely on predictors. The fact that we are less likely to classify items to a positive class *when they are unlabeled* is understandable when one realises that unlabeled class contains relatively *less* positive observations than the general population.

We calculate the Bayes risk for such scenario and bound the excess risk for an classification based solely on predictors, what sheds light on advantage of using labeling information. Also we introduce empirical Bayes classifiers taking advantage of recent proposals for posterior probability estimators in this context. We show that the variant based on variational autoencoder designed for PU data introduced in Chapter 4 works promisingly when accuracy relative to unlabeled data is considered as an evaluation metric.

6.2 Augmented PU prediction method and its properties

We consider now augmented prediction for PU observations (augmented PU prediction) scenario when a new observation (X,S) is given and we want to predict the corresponding value of Y. Obviously, when $S=1$ under assumed scenario we have $Y=1$ and thus we need to consider only the case $S=0$. We introduce the following prediction rule

$$d_B^{PU}(x,s)=\begin{cases}1, & \text{if } s=1\\ \begin{cases}1, & \text{if } y(x)>\frac{1+s(x)}{2}\\ 0, & \text{otherwise,}\end{cases} & \text{if } s=0\end{cases} \tag{6.1}$$

where $y(x)$ is posterior probability of positive class. We will investigate the loss of efficiency when label S, which carries information about Y, is not available for classification. To this end we consider Bayes rule $d_B(x)$ based solely on x:

$$d_B(x)=\begin{cases}1, & \text{if } y(x)>\frac{1}{2}\\ 0, & \text{otherwise.}\end{cases} \tag{6.2}$$

Directly from the above definitions we have that $d_B^{PU}(X,S)$ is more conservative on class $S=0$ than $d_B(X)$ i.e. it less likely assigns objects to the positive class:

$$P(d_B^{PU}(X,S)=1|S=0)\le P(d_B(X)=1|S=0).$$

Below we show that the rule d_B^{PU} is optimal for 0–1 loss and calculate its risk and the excess risk of $d_B(x)$. The fact that the optimal rule is less likely to assign positive class to unlabeled observations than d_B is due to the fact that positive observations occur less frequently among unlabeled ones than in general population. Note that as d_B^{PU} is more conservative, classification changes might occur for both positive and negative examples – though in the expectation, the precision gain should outweigh the lost recall. Also, it has practical consequences for recommendations on the thresholds applied for classification in the follow-up studies involving PU data; see Section 6.4. The introduced approach is based on a simple observation that the considered problem can be regarded as a problem of determining the Bayes risk in the case when the vector of predictors is *augmented* by an additional predictor S. This also motivates the name of the problem. We let

$$\widetilde{y}(x,s)=P(Y=1|(X,S)=(x,s)) \tag{6.3}$$

be posterior probability of $Y=1$ given the augmented vector of predictors. We also define the excess risk (or regret) of any augmented PU prediction rule $d(x,s)$ as (see e.g. Menon et al. (2015)):

$$\Delta(d)=P\left(d(X,S)\neq Y\right)-P\left(d_B^{PU}(X,S)\neq Y\right).$$

Theorem 6.1. *(i) $d_B^{PU}(X,S)$ defined in (6.1) is the Bayes rule for Y under $P_{X,Y,S}$ i.e. it is the classification rule yielding the smallest misclassification error $P(d(X,S)\neq Y)$. Moreover, $d_B^{PU}(X,0)$ is the Bayes rule for Y under $P_{X,Y|S=0}$ yielding the smallest classification error $P(d(X)\neq Y|S=0)$.*

(ii) Define $w(x)=1+s(x)-2y(x)$. Then Bayes risk of $d_B^{PU}(x,s)$ equals

$$\begin{aligned} L_{PU}^* &= \frac{1}{2}\Big(P(S=0)-\mathbb{E}_{X,S=0}\,|2\widetilde{y}(X,0)-1|\Big)\\ &= \frac{1}{2}\Big(P(S=0)-\mathbb{E}_X\,|w(X)|\Big)\end{aligned} \tag{6.4}$$

(iii) We have for excess risk of $d_B(x)$:

$$\mathbb{E}_X\left(s(X)\mathbb{I}\left\{y(X)<\frac{1}{2}\right\}\right)\le\Delta(d_B)\le P(S=1). \tag{6.5}$$

The inequalities above are tight when $P\left(y(X)<\frac{1}{2}\right)=1$.

(iv) *Odds ratio $OR(x)$ for odds of $Y = 1$ in class $\{S = 0\}$ and odds of $Y = 1$ in a general population equals*

$$OR(x) = \frac{P(Y=1|S=0,X=x)}{P(Y=-1|S=0,X=x)} \Big/ \frac{P(Y=1|X=x)}{P(Y=-1|X=x)} \\ = 1 - e(x). \tag{6.6}$$

Proof. (i) As we have

$$P\Big(d_B^{PU}(X,S) \neq Y\Big) = P\Big(d_B^{PU}(X,S) \neq Y \Big| S=1\Big) P(S=1) \\ + P\Big(d_B^{PU}(X,S) \neq Y \Big| S=0\Big) P(S=0)$$

and the first term on RHS equals 0, it is enough to prove that the second and the third line in (6.1) define Bayes rule on the strata $\{S = 0\}$. The Bayes rule for this problem is given by assigning $Y = 1$ when the following condition holds:

$$\frac{P(Y=1|S=0,X=x)}{P(Y=-1|S=0,X=x)} > 1.$$

Denoting by $f(x)$ either the density of X or its probability mass function at x we have, inverting conditional probabilities, that that the ratio above equals

$$\frac{P(S=0,Y=1,X=x)}{P(S=0,Y=-1,X=x)} = \frac{f(x)(y(x)-s(x))}{f(x)(1-y(x))} \\ = \frac{y(x)-s(x)}{1-y(x)}. \tag{6.7}$$

Then it is enough to note that

$$\frac{y(x)-s(x)}{1-y(x)} > 1 \quad \equiv \quad y(x) > \frac{1+s(x)}{2}. \tag{6.8}$$

(ii) As $d_B^{PU}(x,s)$ is Bayes classifier, its risk equals

$$L_{PU}^* = \mathbb{E}_{X,S} \min\big(\widetilde{y}(X,S), 1-\widetilde{y}(X,S)\big), \tag{6.9}$$

where $\widetilde{y}(x,s)$ is defined in (6.3). This is easily justified by noting that if $\widetilde{y}(x,s) > \frac{1}{2}$ and thus (x,s) is assigned to a positive class by the Bayes classifier, it commits an error with probability $1-\widetilde{y}(x,s) = \min(\widetilde{y}(x,s), 1-\widetilde{y}(x,s))$. Moreover, we have that $\widetilde{y}(x,1) = 1$ and reasoning as in (6.7) we obtain

$$\widetilde{y}(x,0) = P(Y=1|(X,S)=(x,0)) \\ = \big(y(x)-s(x)\big) / \big(1-s(x)\big).$$

In view of $\min(a, b) = (a + b - |b - a|)/2$ we have $\min(a, 1-a) = (1 - |2a - 1|)/2$ and whence (6.9) implies that

$$\begin{aligned} L^*_{PU} &= \frac{1}{2} - \frac{1}{2}\,\mathbb{E}_{X,S}\left|2\widetilde{y}(X,S) - 1\right| \\ &= \frac{1}{2} - \frac{1}{2}\,\mathbb{E}_{X,S=1}\left|2\widetilde{y}(X,1) - 1\right| \\ &\quad - \frac{1}{2}\,\mathbb{E}_{X,S=0}\left|2\widetilde{y}(X,0) - 1\right| \\ &= \frac{1}{2} - \frac{1}{2}P(S=1) - \frac{1}{2}\,\mathbb{E}_{X,S=0}\left|2\widetilde{y}(X,0) - 1\right|. \end{aligned} \tag{6.10}$$

Thus we established the first equality in (6.4). Noting that

$$\begin{aligned} &\mathbb{E}_{X,S=0}\left|2\widetilde{y}(X,0) - 1\right| \\ &= \int \frac{\left|2y(x) - s(x) - 1\right|}{1 - s(x)} f(x)(1 - s(x))\,\mathrm{d}x \\ &= \mathbb{E}_X\,|w(X)| \end{aligned}$$

we establish the second one. We note that from the proof above it follows that $d_B^{PU}(x, 0)$ is the Bayes classifier on the strata $\{S = 0\}$ and its Bayes risk L^{*0}_{PU} equals

$$\begin{aligned} L^{*0}_{PU} = \frac{L^*_{PU}}{P(S=0)} &= \frac{1}{2} - \frac{1}{2}\,\mathbb{E}_{X|S=0}\left|2\widetilde{y}(X,0) - 1\right| \\ &= \frac{1}{2} - \frac{\mathbb{E}_X\,|w(x)|}{P(S=0)}. \end{aligned} \tag{6.11}$$

(iii) Reasoning as above we have

$$L^* = P(d_B(X) \neq Y) = \frac{1}{2} - \frac{1}{2}\,\mathbb{E}_X\left|2y(X) - 1\right|$$

and in view of (6.10) we obtain

$$L^* - L^*_{PU} = \frac{1}{2}P(S=1) \quad + \frac{1}{2}\,\mathbb{E}_X\left\{\left|2y(X) - s(X) - 1\right| - \left|2y(X) - 1\right|\right\}.$$

RHS of (6.5) is obtained by using triangle inequality $\left|2y(X) - s(X) - 1\right| - \left|2y(X) - 1\right| \leq s(x)$. To prove LHS of (6.5) we note that we have the following refinement of triangle inequality for $b \geq 0$

$$|a - b| \geq |a| - b + 2b \times \mathbb{I}\{a < 0\}$$

Applying this to $a := 2y(X)-1$ and $b := s(X)$ we have that

$$\left|2y(X)-s(X)-1\right| \geq \left|2y(X)-1\right| - s(x) + 2s(X)\mathbb{I}\left\{y(X) < \frac{1}{2}\right\}$$

and this implies the conclusion as $\mathbb{E}s(X) = P(S=1)$. Note that the lower bound equals the upper bound when for all x we have $y(x) < \frac{1}{2}$. In this case we note that $P\big(d_B(X) \neq Y\big) = P(Y=1)$ whereas $P\big(d_B^{PU}(X,S) \neq Y\big) = P(S=0, Y=1)$ and the excess risk is thus $P(Y=1) - P(S=0,Y=1) = P(S=1)$. The result in (iii) is intuitive: d_B^{PU} does not err on $S=1$, whereas d_B commits an error on this stratum if $y(x) < 1/2$.

(iv) This follows by noting that in view of above derivations

$$\mathrm{OR}(x) = \frac{y(x)-s(x)}{1-y(x)} \Big/ \frac{y(x)}{1-y(x)} = \frac{y(x)-s(x)}{y(x)} = 1-e(x).$$

□

Remark 6.2. *(i) In the view of Lemma 2.1, the threshold in (6.1) can be expressed as*

$$y(x) > \frac{1+s(x)}{2} \equiv y(x) > \frac{1}{2-e(x)}.$$

When $e(x)$ is large, then unlabeled element is less likely to be positive and the threshold becomes larger.

(ii) We note that when labeling is independent of an object in a positive class (SCAR assumption) and thus propensity score $e(x) \equiv c$, we have (cf (i)):

$$d_B^{PU}(x,0) = 1 \iff y(x) > \frac{1}{2-c}.$$

For situation of complete lack of labeling ($c=0$) unlabeled class is distributed according to P_X and $d_B^{PU}(x,0)$ coincides with $d_B(x)$ in agreement with the last inequality. Note that since under SCAR positive observations are labeled or not, regardless of the predictors' values, the threshold $(2-c)^{-1}$ above is due solely to the changed proportion of positives among unlabeled ones compared with the general population.

Below we calculate excess risk in (6.5) for a specific model.

Example 6.3. *Let $y(x) = \Phi(x), X \sim \mathcal{N}(0,1)$, and $x \in \mathbb{R}$ (univariate probit model with standard normal predictor), and let propensity score $e_a(x) = \mathbb{I}\{x > a\}$ i.e. above threshold $a \in \mathbb{R}$ all positive observations are labeled. In this case the excess risk of $d_B(x)$ defined in (6.2) for $a > 0$ equals (refer*

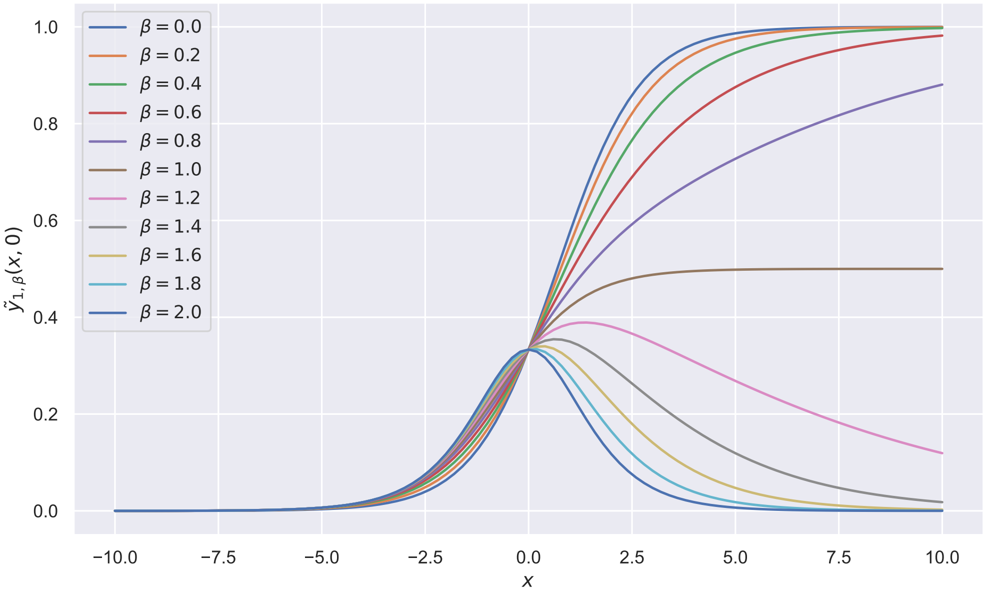


Figure 6.1: Values of $\widetilde{y}_{1,\beta}(x,0)$ depending on β.

to appendix C for full derivation)

$$\begin{aligned}\Delta(d_B) &= \mathbb{E}_X\Big[\min\big(y(X), 1-y(X)\big)\Big] \\ &\quad -\mathbb{E}_{X,S}\Big[\min\big(\widetilde{y}(X,S), 1-\widetilde{y}(X,S)\big)\Big] \\ &= \frac{1}{2} - \Phi(a) + \frac{\Phi^2(a)}{2} = \frac{1}{2}\big(\Phi(a)-1\big)^2 \geq 0,\end{aligned}$$

and for $a < 0$ equals $\frac{1}{4} - \frac{\Phi^2(a)}{2} \geq 0$. Note that for $a \to \infty$ excess risk tends to 0 as $P_{X,S=0}$ approaches P_X in this case and $d_B^{PU}(x,0)$ tends to $d_B(x)$. For $a \to -\infty$ the excess risk tends to 1/4 (risk of $d_B(x)$) as the risk of $d_B^{PU}(x,s)$ tendto 0 0.

Example 6.4. *Consider the situation when $y(x) = \sigma(\alpha x)$ and $e(x) = \sigma(\beta x)$ for $x \in \mathbb{R}$ and $\alpha, \beta \geq 0$. Then we have for $\widetilde{y}(x,0)$ defined in (6.3)*

$$\begin{aligned}\widetilde{y}_{\alpha,\beta}(x,0) &= \frac{y(x)-s(x)}{1-s(x)} = \frac{\sigma(\alpha x) - \sigma(\alpha x)\sigma(\beta x)}{1-\sigma(\alpha x)\sigma(\beta x)} \\ &= \frac{\frac{1}{\sigma(\beta x)} - 1}{\frac{1}{\sigma(\alpha x)\sigma(\beta x)} - 1} = \frac{1}{1+e^{-(\alpha-\beta)x} + e^{-\alpha x}}.\end{aligned} \tag{6.12}$$

The plot of $\widetilde{y}_{\alpha,\beta}(x,0)$ for $\alpha = 1$ and various βs is shown on Figure 6.1. Note that for $\alpha = \beta$ we have $\widetilde{y}_{\alpha,\alpha}(x,0) = (2+\exp(-\alpha x))^{-1}$ which tends to $\frac{1}{2}$ when $x \to +\infty$, indicating the most difficult situation when $\widetilde{y}(x,0)$ is in a vicinity of $\frac{1}{2}$.

6.3 d_B^{PU} applications – VAE-PU-Bayes

The proposed d_B^{PU} rule uses are not limited to the direct applications to the augmented PU prediction style data (where the observation label is available for the test data). As a motivational example we consider first a typical PU problem, with only predictors available at the test time.

VAE-PU-Bayes is another novel proposal which aims to further improve upon performance of VAE-PU+OCC proposed in Chapter 4. Inner selection of the predicted positive examples is a crucial part of the VAE-PU+OCC, but general purpose one-class classifiers are – on the whole – a relatively low power methods, as they work with very limited information – only using the inlier example distribution. Note that even in the standard PU problem, we can use more information than that, as we have access label information for all of the training examples. This allows for training a classifier which can be used for $s(x)$ estimation. VAE-PU-Bayes, which we introduce here, combines such a classifier with the VAE-PU $y(x)$ estimation in order to apply d_B^{PU} rule. As here the aim is to only filter the unlabeled set, we have $S = 0$ on it and can use the relevant part of the d_B^{PU} rule. Note that for the unlabeled stratum, we can rewrite it as follows (see Eq. (6.8)):

$$y(x) > \frac{1+s(x)}{2} \equiv \frac{y(x)-s(x)}{1-y(x)} > 1.$$

An important consideration is that for numerical reasons, the proportions in the training dataset are crucial to VAE-PU training (recall Section 4.5.3 for details) – due to that, the number of selected likely positives should reflect the true portion on unlabeled positives present in the dataset. Thus, instead of choosing all unlabeled elements satisfying $\frac{y(x)-s(x)}{1-y(x)} > 1$ as likely positives, we calculate an example's score as $\frac{y(x)-s(x)}{1-y(x)}$ and select the appropriate number (as defined in Algorithm 2, step 7) of examples with the highest score as an approximation of internal true PU set. This approach to PU set generation is consistent with the decision rule proposed in the chapter, and can significantly outperform OCC-based models due to more powerful classification approach.

6.4 Numerical experiments

To check the effectiveness of the proposed approach, we prepared an extensive suite of experiments. We considered 4 synthetic and 6 real-world datasets:

- All synthetic datasets are generated using a mixture of two 20D Gaussian distributions (with different means 0 and μ and unit covariance I, except Variant 3) as a feature vector. This implies that indicator Y of an element of a mixture is drawn from the logistic distribution with parameter β (β is equal to direction of LDA boundary between feature clusters; we use intercept value which ensures $\pi = 0.5$). The following variants were used:

  - **Variant 1.** Propensity score for a positive example $e_1(x)$ equals $\sigma(\gamma^T x + r)$, $\sigma(\cdot)$ being the logistic function, parameter vector $\gamma = [\gamma_1, \gamma_2, ..., \gamma_p] = [0.5, 0.5, ..., 0.5]$ and intercept r is tuned to ensure correct label frequency. Intercept tuning uses the assumed label frequency error $|\widetilde{c} - c|$ (absolute difference between empirical label frequency in the dataset and its target value) as the objective, which is minimized using differential evolution algorithm.This allows us to construct synthetic datasets with both required labeling probabilities and label frequencies.
  - **Variant 2.** Propensity score: $e_2(x) = e_1(x)^{10}$ which approximates step-wise function and has been considered in Gong et al. (2021).
  - **Variant 3.** In this variant, covariance matrix is diagonal, non-unit matrix in order to obtain non-logistic data (the diagonal vector equals: $[1, 2, 1, 2, ..., 1, 2]$), $e_3(x) = e_1(x)$,
  - **Variant 4 (SCAR).** Constant propensity score, equal to label frequency: $e_4(x) = c$ (equivalent to the SCAR assumption).

- Real-world datasets, with characteristics of the datasets and their labeling as given in the Appendix E. Here we keep dataset naming convention from the previous chapter.

We propose the following variants of the three popular no-SCAR PU methods:

- **LBE+S**. LBE (Gong et al. 2021) method is a natural candidate due to explicit modeling of both posterior probability $y(x)$ of $Y = 1$ and propensity score $e(x)$ (recall that we can obtain posterior probability of $S = 1$ by using $s(x) = e(x)y(x)$). After training the LBE classifier, we use both fitted components as plug-in estimators of $y(x)$ and $s(x)$ values in d_B^{PU} rule.

- **VAE-PU+S** (abbrev. **VP+S**). We use VAE-PU[1] (Chapter 4.) classifier as the base. As this model does not natively use the notion of propensity score in contrast to LBE, we introduce a separate feed-forward neural network for $s(x)$ estimation, trained separately from VAE-PU in the additional training step. Its predictions are then fed (together with VAE-PU's $y(x)$ estimations) to the proposed decision rule.

- **VAE-PU-Bayes+S** (abbrev. **VP-B+S**). We use a newly introduced VAE-PU-Bayes classifier (described in Section 6.3) as the base. Similarly to VAE-PU, $s(x)$ estimator is trained and provided externally using available $(X_i, S_i)_{i=1}^n$ set.

Note that for synthetic datasets, we can obtain accurate values of both $y(x)$ and $s(x)$; for those datasets we will additionally show results of the following two pseudo-methods:

[1] VAE-PU implementation includes the baseline modifications described in Chapter 4

- **S-Prophet**. Corresponds to the application of d_B^{PU} rule (6.1) with exact $y(x)$ and $s(x)$.
- **Y-Prophet**. Corresponds to a "naive" approach, where a researcher infers $Y = 1$ for test labeled examples with $S = 1$; but then (as one would in the standard PU task) blindly applies (6.2) to all other examples. Note that we assume knowledge of $y(x)$.

We also define a "naive" versions of LBE+S, VAE-PU+S and VAE-PU-Bayes+S in a similar way (as LBE, VAE-PU and VAE-PU-Bayes) – by assuming $Y = 1$ for labeled test examples, and using the simple d_B rule for the unlabeled examples.

In order to evaluate the performance, we focus on the "U-metrics", that is metrics calculated for unlabeled stratum. As prediction for labeled test examples is trivial, omitting them in the evaluation results paints clearer picture of the true, underlying decision performance. As an example, U-Accuracy is an Accuracy calculated only on the $S = 0$ stratum: $U\text{-}ACC = {n_U}^{-1} \sum_{x_U \in U} \mathbb{I}\{d(x_U, s) = y_U\}$.

We prove the effectiveness of the proposed modification in two steps. First, we show which of the proposed variants (relying on d_B^{PU} rule) performs the best on our benchmark tasks. We then go on to compare the best variant with its naive counterpart, showing the benefits of applying our proposed decision rule. Each experiment (defined as a combination of dataset, label frequency and method) was performed 10 times, each time initialized with a different random seed (equal to experiment number). All code used for method implementation and performed experiments is publicly available at GitHub[2].

The result section will also contain a brief comparison of VAE-PU-Bayes (abbrev. VP-B) method with the baseline VAE-PU (abbrev. VP; it includes modifications described in Chapter 4) and two recommended VAE-PU+OCC variants – A^3 (abbrev. VP-$\boldsymbol{A^3}$) and Isolation Forest (abbrev. VP-**IF**). Those experiments were performed without test label availability, and use accuracy as the main metric. The other experimental settings do not differ from the augmented PU prediction experiments. The code for this method is a modification of the original VAE-PU+OCC code, also publicly available in a separate GitHub repository[3].

6.4.1 Results of experiments

VAE-PU-Bayes. First, we show the effectiveness of VAE-PU-Bayes in traditional PU setting. Table 6.1 presents the accuracy comparison between the newly introduced variant and previously existing VAE-PU and VAE-PU+OCC. In the vast majority of cases it outperforms the other VAE-PU variants, often by a very large margin – up to 5 percentage points (pp.), as in the case of CDC-Diabetes. The only exceptions are the lowest label frequency $c = 0.02$, where is it

[2] `https://github.com/wawrzenczyka/VP-Bayes-S`

[3] `https://github.com/wawrzenczyka/VAE-PU-Bayes`

Table 6.1: Accuracy values – VAE-PU-Bayes (traditional PU setting)

c	Method	MNIST 3v5	MNIST OvE	CIFAR CT	CIFAR VA	STL VA	CDC Diabetes
0.02	VP	**79.67 ± 0.90**	70.00 ± 1.76	87.31 ± 0.58	90.51 ± 0.52	81.64 ± 0.44	50.82 ± 0.22
	VP-A^3	79.01 ± 0.70	74.89 ± 1.62	83.67 ± 1.05	90.73 ± 0.27	79.62 ± 0.55	53.77 ± 1.33
	VP-IF	79.07 ± 0.75	**76.87 ± 0.99**	89.98 ± 1.23	89.99 ± 0.36	79.98 ± 1.06	52.38 ± 1.20
	VP-B	78.65 ± 0.87	73.13 ± 1.70	**91.74 ± 0.90**	**93.70 ± 0.17**	**84.68 ± 0.65**	**57.92 ± 1.31**
0.10	VP	83.57 ± 0.59	77.08 ± 0.92	91.22 ± 0.19	91.54 ± 0.31	85.31 ± 0.32	51.37 ± 0.25
	VP-A^3	89.91 ± 0.32	83.14 ± 1.41	90.35 ± 0.47	92.35 ± 0.28	84.91 ± 0.44	59.78 ± 1.36
	VP-IF	90.12 ± 0.30	83.60 ± 1.28	92.26 ± 0.39	90.21 ± 0.57	85.52 ± 0.56	57.24 ± 1.51
	VP-B	**90.76 ± 0.54**	**85.72 ± 1.05**	**93.73 ± 0.17**	**94.39 ± 0.14**	**88.44 ± 0.28**	**63.46 ± 0.83**
0.30	VP	86.77 ± 0.48	83.71 ± 0.27	92.96 ± 0.28	93.72 ± 0.13	88.23 ± 0.27	54.32 ± 0.26
	VP-A^3	92.65 ± 0.22	90.49 ± 0.23	89.88 ± 0.68	93.45 ± 0.12	86.38 ± 0.37	68.32 ± 0.38
	VP-IF	92.73 ± 0.22	90.59 ± 0.23	93.37 ± 0.21	92.02 ± 0.44	87.11 ± 0.51	67.72 ± 0.44
	VP-B	**92.98 ± 0.34**	**90.88 ± 0.27**	**94.22 ± 0.13**	**94.95 ± 0.06**	**89.99 ± 0.26**	**69.87 ± 0.19**
0.50	VP	88.32 ± 0.55	80.87 ± 1.35	92.91 ± 0.31	88.19 ± 0.37	88.57 ± 0.49	60.58 ± 0.37
	VP-A^3	93.28 ± 0.59	91.88 ± 0.42	88.03 ± 1.16	93.44 ± 0.12	87.46 ± 0.24	70.97 ± 0.19
	VP-IF	93.40 ± 0.55	91.59 ± 0.35	93.74 ± 0.13	92.04 ± 0.26	87.91 ± 0.35	70.66 ± 0.22
	VP-B	**93.92 ± 0.38**	**92.10 ± 0.28**	**94.46 ± 0.17**	**94.99 ± 0.05**	**90.57 ± 0.30**	**71.79 ± 0.12**
0.70	VP	91.58 ± 0.60	91.17 ± 0.29	94.20 ± 0.20	94.67 ± 0.08	90.12 ± 0.34	65.91 ± 0.25
	VP-A^3	93.89 ± 0.46	94.10 ± 0.28	88.93 ± 1.41	93.99 ± 0.07	89.06 ± 0.28	72.01 ± 0.07
	VP-IF	94.21 ± 0.39	94.39 ± 0.25	93.99 ± 0.16	93.74 ± 0.10	89.27 ± 0.32	71.93 ± 0.15
	VP-B	**94.59 ± 0.57**	**94.78 ± 0.16**	**94.51 ± 0.18**	**95.28 ± 0.04**	**91.01 ± 0.24**	**72.42 ± 0.07**
0.90	VP	94.63 ± 0.17	93.15 ± 0.25	**94.49 ± 0.14**	94.79 ± 0.13	91.14 ± 0.23	71.02 ± 0.17
	VP-A^3	95.35 ± 0.15	95.90 ± 0.10	91.12 ± 0.34	94.69 ± 0.09	91.03 ± 0.24	72.21 ± 0.07
	VP-IF	**95.70 ± 0.18**	95.80 ± 0.12	94.48 ± 0.19	94.58 ± 0.08	**91.29 ± 0.29**	**72.45 ± 0.13**
	VP-B	95.29 ± 0.16	**95.96 ± 0.11**	94.29 ± 0.22	**95.12 ± 0.13**	91.16 ± 0.27	72.20 ± 0.13

outperformed on MNIST datasets, and $c = 0.9$, but even in this last case it is roughly comparable to the best alternative. The original VAE-PU achieved the best performance only in 2 cases out of 36, with a very small margin. As VAE-PU+OCC was shown to achieve state-of-the-art level performance when compared to non-generative alternatives (see Chapter 4.8.1), VAE-PU-Bayes can be recommended as an improved variant of this model for traditional PU learning problems.

Augmented PU prediction. The rest of the result Section focuses on augmented PU prediction scenario (with available test labels). We stress that the aim here is to choose the best performing method among possible proposals for the new scenario. Tables 6.2 and 6.3 aggregate experiments performed with d_B^{PU} rule for synthetic and real-world datasets, respectively. The best U-Accuracy is marked in bold for each dataset and label frequency combination. The results for Balanced Accuracy are given in the Appendix D (note that the ratio of positives to negatives among unlabeled equals $\pi(1-c)/(1-\pi)$ and may be small for $c = 0.7, 0.9$). For synthetic datasets, VP-B+S is the top performer in the low frequency region; LBE+S does not work well for low label frequencies, but tends to overtake VP-B+S for $c = 0.7$ – then it levels off and falls off for $c = 0.9$. Even though VP+S is better that VP-B+S for $c = 0.9$, it is outperformed by it for all other label frequencies. For real-world datasets, VP-B+S shows even better performance, dominating in the vast majority of test cases, except for high label frequencies $c = 0.5, 0.7$ in the case of CDC

Table 6.2: U-Accuracy values – Method comparison – Synthetic datasets

c	Method	Synth. 1	Synth. 2	Synth. 3	Synth. SCAR
0.02	S-Prophet	73.29 ± 0.35	73.24 ± 0.35	71.37 ± 0.35	73.48 ± 0.35
	VP+S	60.55 ± 2.48	59.15 ± 2.62	59.77 ± 2.40	63.19 ± 1.75
	VP-B+S	**61.23 ± 2.35**	**59.35 ± 2.66**	**60.16 ± 2.36**	**63.45 ± 1.82**
	LBE+S	50.32 ± 0.50	50.59 ± 0.50	50.66 ± 0.49	50.29 ± 0.47
0.10	S-Prophet	72.63 ± 0.30	72.16 ± 0.35	70.61 ± 0.30	73.74 ± 0.34
	VP+S	67.18 ± 0.42	65.96 ± 0.58	67.02 ± 0.57	67.64 ± 0.42
	VP-B+S	**67.71 ± 0.49**	**66.63 ± 0.60**	**67.49 ± 0.59**	**68.37 ± 0.50**
	LBE+S	52.72 ± 0.47	53.45 ± 0.50	53.04 ± 0.45	52.39 ± 0.53
0.30	S-Prophet	71.70 ± 0.42	70.83 ± 0.48	69.45 ± 0.39	74.30 ± 0.46
	VP+S	67.77 ± 0.57	65.29 ± 0.64	66.90 ± 0.55	70.20 ± 0.45
	VP-B+S	**68.51 ± 0.54**	**66.41 ± 0.57**	**67.27 ± 0.47**	**71.03 ± 0.42**
	LBE+S	61.05 ± 0.36	60.80 ± 0.43	61.03 ± 0.31	58.80 ± 0.52
0.50	S-Prophet	72.78 ± 0.57	71.96 ± 0.46	70.75 ± 0.59	76.93 ± 0.56
	VP+S	66.87 ± 0.41	65.04 ± 0.47	66.07 ± 0.67	69.78 ± 0.68
	VP-B+S	67.90 ± 0.45	65.57 ± 0.46	67.01 ± 0.51	**72.31 ± 0.38**
	LBE+S	**68.72 ± 0.51**	**67.72 ± 0.50**	**68.45 ± 0.45**	70.86 ± 0.48
0.70	S-Prophet	78.79 ± 0.40	78.37 ± 0.35	77.70 ± 0.50	81.31 ± 0.37
	VP+S	67.28 ± 0.88	66.38 ± 0.88	66.05 ± 0.78	69.34 ± 1.27
	VP-B+S	70.49 ± 0.54	68.91 ± 0.51	69.04 ± 0.48	73.57 ± 0.59
	LBE+S	**74.74 ± 0.42**	**73.50 ± 0.52**	**73.57 ± 0.42**	**81.03 ± 0.38**
0.90	S-Prophet	91.20 ± 0.49	91.26 ± 0.50	91.42 ± 0.44	91.83 ± 0.36
	VP+S	**85.54 ± 0.49**	**86.00 ± 0.76**	**85.64 ± 0.70**	**87.97 ± 0.55**
	VP-B+S	84.00 ± 0.42	84.48 ± 0.69	83.90 ± 0.56	86.50 ± 0.54
	LBE+S	74.76 ± 0.46	74.53 ± 0.47	73.57 ± 0.55	78.14 ± 0.48

Table 6.3: U-Accuracy values – Method comparison – Real-world datasets

c	Method	MNIST 3v5	MNIST OvE	CIFAR CT	CIFAR VA	STL VA	CDC-Diabetes
0.02	VP+S	77.74 ± 1.10	68.47 ± 1.18	87.19 ± 0.49	90.32 ± 0.25	81.43 ± 0.66	49.76 ± 1.49
	VP-B+S	**78.74 ± 1.53**	**74.91 ± 1.51**	**92.45 ± 0.43**	**94.11 ± 0.09**	**84.54 ± 0.67**	**51.15 ± 1.74**
	LBE+S	47.38 ± 0.32	49.82 ± 0.14	50.50 ± 0.40	60.67 ± 0.22	60.55 ± 0.28	50.47 ± 0.20
0.10	VP+S	80.21 ± 0.61	73.96 ± 1.43	91.42 ± 0.33	91.93 ± 0.33	86.36 ± 0.38	56.57 ± 0.79
	VP-B+S	**84.32 ± 0.76**	**83.12 ± 1.24**	**93.45 ± 0.19**	**94.37 ± 0.12**	**88.74 ± 0.30**	**61.01 ± 0.69**
	LBE+S	49.81 ± 0.34	51.55 ± 0.15	53.19 ± 0.39	62.81 ± 0.29	62.93 ± 0.28	52.55 ± 0.20
0.30	VP+S	80.66 ± 0.73	78.38 ± 0.96	92.95 ± 0.30	93.49 ± 0.16	88.93 ± 0.20	51.76 ± 0.94
	VP-B+S	**86.95 ± 0.52**	**87.98 ± 0.64**	**94.32 ± 0.14**	**95.22 ± 0.07**	**89.90 ± 0.30**	**62.63 ± 0.89**
	LBE+S	56.26 ± 0.34	57.11 ± 0.15	63.07 ± 1.04	74.37 ± 2.36	71.53 ± 1.13	58.72 ± 0.20
0.50	VP+S	82.25 ± 0.78	81.15 ± 0.93	94.39 ± 0.20	94.63 ± 0.26	90.85 ± 0.26	48.01 ± 0.85
	VP-B+S	**89.20 ± 0.81**	**90.25 ± 0.56**	**95.13 ± 0.20**	**95.65 ± 0.12**	**91.44 ± 0.30**	66.88 ± 0.37
	LBE+S	64.23 ± 0.33	64.80 ± 0.23	80.81 ± 1.68	85.35 ± 1.18	84.15 ± 1.00	**72.39 ± 0.81**
0.70	VP+S	86.42 ± 0.65	85.88 ± 0.67	95.32 ± 0.22	95.53 ± 0.18	**93.27 ± 0.29**	42.13 ± 1.05
	VP-B+S	**92.19 ± 0.47**	**92.76 ± 0.43**	**95.73 ± 0.18**	**96.23 ± 0.12**	**93.27 ± 0.21**	71.74 ± 0.49
	LBE+S	74.84 ± 0.40	76.65 ± 0.22	93.86 ± 0.57	95.16 ± 0.25	92.17 ± 0.38	**77.12 ± 0.79**
0.90	VP+S	91.90 ± 0.35	90.91 ± 0.35	**96.65 ± 0.16**	96.76 ± 0.15	**95.83 ± 0.19**	42.56 ± 2.15
	VP-B+S	**94.07 ± 0.25**	**95.73 ± 0.23**	96.47 ± 0.16	**97.11 ± 0.09**	95.37 ± 0.24	**83.53 ± 0.33**
	LBE+S	90.00 ± 0.27	87.03 ± 0.97	94.21 ± 0.48	94.69 ± 1.25	94.01 ± 0.85	71.09 ± 1.63

Table 6.4: U-Accuracy values – Decision rule comparison – Synthetic datasets

c	Method	Synth. 1	Synth. 2	Synth. 3	Synth. SCAR
0.02	S-Prophet	73.29 ± 0.35	73.24 ± 0.35	71.37 ± 0.35	73.48 ± 0.35
	Y-Prophet	73.31 ± 0.36	73.24 ± 0.35	71.40 ± 0.36	73.50 ± 0.35
	VP-B	61.00 ± 2.40	**59.62 ± 2.56**	60.14 ± 2.38	63.37 ± 1.77
	VP-B+S	**61.23 ± 2.35**	59.35 ± 2.66	**60.16 ± 2.36**	**63.45 ± 1.82**
	VP-B+S + true s(x)	60.65 ± 2.41	59.15 ± 2.69	59.98 ± 2.39	63.14 ± 1.76
	VP-B+S + true y(x)	73.29 ± 0.37	73.21 ± 0.35	71.44 ± 0.35	73.46 ± 0.33
0.10	S-Prophet	72.63 ± 0.30	72.16 ± 0.35	70.61 ± 0.30	73.74 ± 0.34
	Y-Prophet	72.63 ± 0.35	72.19 ± 0.37	70.68 ± 0.37	73.66 ± 0.33
	VP-B	**67.81 ± 0.48**	66.42 ± 0.53	67.38 ± 0.60	68.35 ± 0.48
	VP-B+S	67.71 ± 0.49	**66.63 ± 0.60**	**67.49 ± 0.59**	**68.37 ± 0.50**
	VP-B+S + true s(x)	67.64 ± 0.50	66.33 ± 0.51	67.16 ± 0.62	68.07 ± 0.48
	VP-B+S + true y(x)	72.71 ± 0.34	71.92 ± 0.39	70.61 ± 0.35	73.73 ± 0.32
0.30	S-Prophet	71.70 ± 0.42	70.83 ± 0.48	69.45 ± 0.39	74.30 ± 0.46
	Y-Prophet	71.06 ± 0.39	70.08 ± 0.39	69.00 ± 0.37	73.56 ± 0.34
	VP-B	68.25 ± 0.47	66.27 ± 0.62	67.14 ± 0.52	70.56 ± 0.43
	VP-B+S	**68.51 ± 0.54**	**66.41 ± 0.57**	**67.27 ± 0.47**	**71.03 ± 0.42**
	VP-B+S + true s(x)	68.19 ± 0.54	66.02 ± 0.63	67.06 ± 0.51	70.72 ± 0.44
	VP-B+S + true y(x)	71.26 ± 0.46	70.56 ± 0.52	69.19 ± 0.43	74.32 ± 0.48
0.50	S-Prophet	72.78 ± 0.57	71.96 ± 0.46	70.75 ± 0.59	76.93 ± 0.56
	Y-Prophet	69.87 ± 0.40	68.83 ± 0.39	67.81 ± 0.43	73.26 ± 0.35
	VP-B	67.07 ± 0.40	65.18 ± 0.45	66.14 ± 0.65	70.43 ± 0.46
	VP-B+S	**67.90 ± 0.45**	**65.57 ± 0.46**	**67.01 ± 0.51**	**72.31 ± 0.38**
	VP-B+S + true s(x)	67.58 ± 0.38	65.36 ± 0.51	66.67 ± 0.63	71.99 ± 0.46
	VP-B+S + true y(x)	72.06 ± 0.59	71.11 ± 0.61	69.83 ± 0.62	76.34 ± 0.62
0.70	S-Prophet	78.79 ± 0.40	78.37 ± 0.35	77.70 ± 0.50	81.31 ± 0.37
	Y-Prophet	69.39 ± 0.41	68.79 ± 0.43	67.44 ± 0.47	73.42 ± 0.35
	VP-B	66.46 ± 0.59	65.69 ± 0.56	65.25 ± 0.65	68.76 ± 0.71
	VP-B+S	**70.49 ± 0.54**	**68.91 ± 0.51**	**69.04 ± 0.48**	**73.57 ± 0.59**
	VP-B+S + true s(x)	69.95 ± 0.58	69.16 ± 0.49	68.88 ± 0.53	73.09 ± 0.58
	VP-B+S + true y(x)	77.42 ± 0.45	77.11 ± 0.39	75.95 ± 0.61	80.46 ± 0.46
0.90	S-Prophet	91.20 ± 0.49	91.26 ± 0.50	91.42 ± 0.44	91.83 ± 0.36
	Y-Prophet	71.30 ± 0.44	71.17 ± 0.45	69.25 ± 0.47	73.33 ± 0.48
	VP-B	69.71 ± 0.37	69.47 ± 0.41	68.16 ± 0.49	71.76 ± 0.42
	VP-B+S	**84.00 ± 0.42**	**84.48 ± 0.69**	**83.90 ± 0.56**	**86.50 ± 0.54**
	VP-B+S + true s(x)	84.12 ± 0.56	83.89 ± 0.55	83.56 ± 0.65	87.03 ± 0.49
	VP-B+S + true y(x)	90.44 ± 0.41	90.69 ± 0.47	89.72 ± 0.37	90.82 ± 0.24

Diabetes (where it is outperformed by LBE-S). Overall, we find that VP-B+S is the empirical variant of rule (6.1) most suited for general recommendation and use, thus we will use it in the further results' presentation. We also note that the dependence of performance on labeling frequency c is much less pronounced here than in classical PU inference. This is due to the fact that large value of c in general translates to a relatively smaller number of positive observations among unlabeled ones.

In Tables 6.4 and 6.5, we aim to capture the impact of using d_B^{PU} rule-based VP-B+S instead of its naive counterpart, VP-B. For synthetic datasets, where we have access to true $y(x)$ and $s(x)$ value, we contrast them with the analogous, reference S-Prophet and Y-Prophet methods. In this case, we also introduce the "semi-Prophet" methods – VP-B+S with true $s(x)$ and VP-B+S with

Table 6.5: U-Accuracy values – Decision rule comparison – Real-world datasets

c	Method	MNIST 3v5	MNIST OvE	CIFAR CT	CIFAR MA	STL MA	CDC-Diabetes
0.02	VP-B	**78.75 ± 1.44**	74.53 ± 1.49	92.40 ± 0.41	93.94 ± 0.10	84.50 ± 0.66	51.07 ± 1.76
	VP-B+S	78.74 ± 1.53	**74.91 ± 1.51**	**92.45 ± 0.43**	**94.11 ± 0.09**	**84.54 ± 0.67**	**51.15 ± 1.74**
0.10	VP-B	84.14 ± 0.65	82.67 ± 1.30	93.32 ± 0.18	94.29 ± 0.12	88.54 ± 0.30	**61.25 ± 0.80**
	VP-B+S	**84.32 ± 0.76**	**83.12 ± 1.24**	**93.45 ± 0.19**	**94.37 ± 0.12**	**88.74 ± 0.30**	61.01 ± 0.69
0.30	VP-B	86.64 ± 0.56	87.89 ± 0.65	94.18 ± 0.17	95.11 ± 0.07	**90.11 ± 0.26**	**63.44 ± 0.81**
	VP-B+S	**86.95 ± 0.52**	**87.98 ± 0.64**	**94.32 ± 0.14**	**95.22 ± 0.07**	89.90 ± 0.30	62.63 ± 0.89
0.50	VP-B	88.75 ± 0.84	90.19 ± 0.54	94.87 ± 0.22	95.44 ± 0.11	91.35 ± 0.25	66.52 ± 0.31
	VP-B+S	**89.20 ± 0.81**	**90.25 ± 0.56**	**95.13 ± 0.20**	**95.65 ± 0.12**	**91.44 ± 0.30**	**66.88 ± 0.37**
0.70	VP-B	91.84 ± 0.48	92.73 ± 0.39	95.30 ± 0.20	95.94 ± 0.10	92.64 ± 0.24	67.73 ± 0.43
	VP-B+S	**92.19 ± 0.47**	**92.76 ± 0.43**	**95.73 ± 0.18**	**96.23 ± 0.12**	**93.27 ± 0.21**	**71.74 ± 0.49**
0.90	VP-B	93.90 ± 0.25	95.45 ± 0.21	95.68 ± 0.16	96.51 ± 0.05	93.54 ± 0.26	68.13 ± 0.34
	VP-B+S	**94.07 ± 0.25**	**95.73 ± 0.23**	**96.47 ± 0.16**	**97.11 ± 0.09**	**95.37 ± 0.24**	**83.53 ± 0.33**

true $y(x)$, where the true values replace the corresponding VP-B+S probability estimation. First thing to note is that S-Prophet is nearly equivalent to Y-Prophet in low label frequency setting. When there is a small number of labeled examples, it leads to low in expectation predicted labeling probability $s(x)$. As the rules d_B^{PU} and d_B are equivalent when $s(x) = 0$, for low label frequencies the change in predicted class is relatively infrequent. As the label frequency increases, so does the discrepancy between prophet methods – culminating in the drastic difference of 20 pp. for $c = 0.9$. The differences between VP-B+S and VP-B are not as big, and also tend to increase jointly with label frequency. However, p-value of the binomial test for testing H_0: P(U-acc. of VP-B $>$ U-acc. of VP-B+S)$\geq 1/2$ against the opposite hypothesis, equals to 1.8×10^{-5} (corresponding to 2 wins in 24 trials) for Table 6.4 and 1.1×10^{-7} in case of Table 6.5. Using the correct decision rule via VP-B+S we obtain U-Accuracy increase in almost every test scenario, though the margin here is much smaller than in the case of Prophets, and more pronounced for synthetic datasets. Inspecting semi-Prophet results gives us additional insights into the d_B^{PU} components. Note that when using true $y(x)$, the VP-B+S semi-Prophet's accuracy does not deviate significantly from S-Prophet's – even though the estimation of $s(x)$ was fairly crude, it is good enough when combined with accurate $y(x)$ estimations to improve results significantly. The same does not hold true for VP-B+S semi-Prophet using true $s(x)$ values, which indicates that $y(x)$ estimation inaccuracy is a major contributor to the performance drop compared with the Prophet methods. This is evident by contrasting the results with the S-Prophet – Y-Prophet pair, where using true $s(x)$ for d_B^{PU} rule proved to increase performance dramatically for high label frequencies. Note that sometimes even a slight variation of $y(x)$ might led to a between d_B^{PU} influencing the final example label or leaving it unchanged.

Results described above show that in real-world scenarios, the performance gain obtained by using the proposed decision rule over d_B rule is systematic but relatively small; this is especially apparent when comparing it to Prophets' improvements. Figure 6.2 aims to illustrate one of the

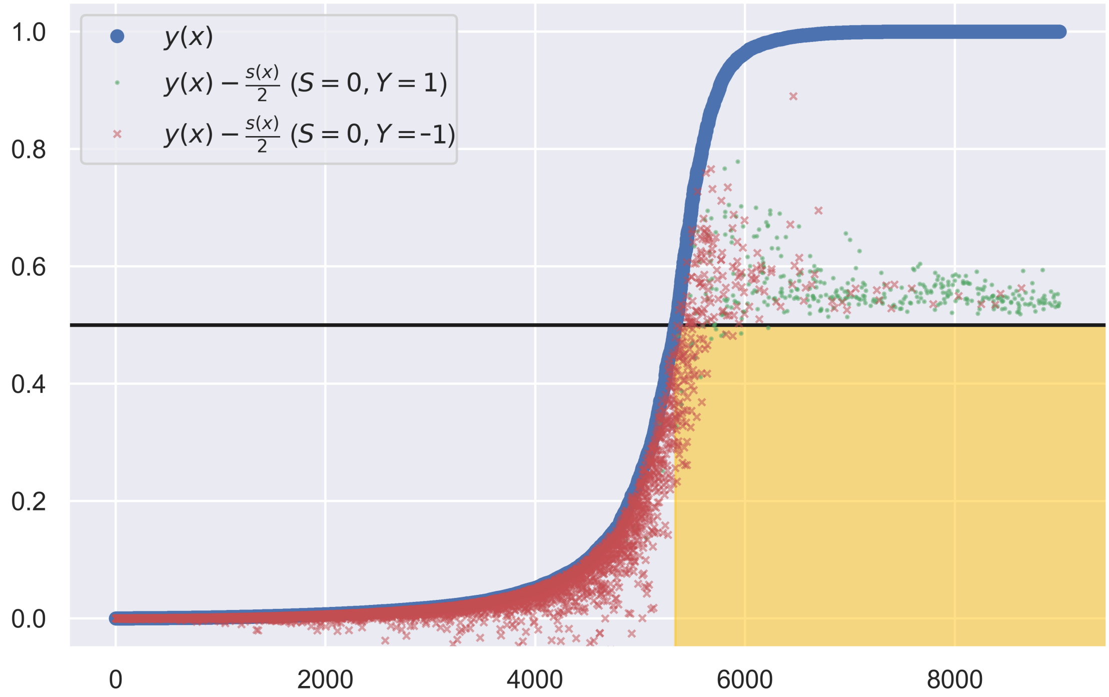


Figure 6.2: Classification rules for test instances, CIFAR VA, $c = 0.9$, $S = 0$ stratum (colored by test class).

potential causes of that problem. For the sake of this example, we will plot those values only for examples from $S = 0$ stratum. Note that for this stratum, d_B rule is equivalent to $\mathbb{I}\{y(x) > 0.5\}$, whereas d_B^{PU} – to $\mathbb{I}\{y(x) - \frac{s(x)}{2} > 0.5\}$. This formulation provides us with a matching threshold 0.5 for both rules.

The example orders the test examples according to increasing $y(x)$ (blue color, basis of d_B rule). In the figure, we introduce one additional dot for each test instance, which now corresponds to $y(x) - \frac{s(x)}{2}$ (basis of d_B^{PU} rule). Those dots are colored based on their test class (positive examples in green, and negative – in red). As $y(x) - \frac{s(x)}{2}$ is always lower or equal to $y(x)$, in order for the d_B and d_B^{PU} classification rules to differ on the $S = 0$ stratum (i) the blue dot for the given test example must be above black boundary line ($y = 0.5$), and (ii) the other dot (green or red) must be lying below it. The area in the chart where this is possible is shaded gold, and only examples falling there fulfill both conditions. The important thing to note is that the amount of examples falling in the golden area is relatively small, due to approximated $y(x)$ tending to the extremes of 0 and 1 – which might not hold true for the true $y(x)$ distribution. This limits the benefits of applying the d_B^{PU} rule, as even though the examples in the golden area are mostly negative (resulting in decreasing of the number of false signals), and the green, positive unlabeled examples are concentrated in the high $y(x)$ area, the limited number of affected examples by rule's modification lowers the impact of the correction on the metrics such as U-Accuracy.

6.5 Conclusions

The contribution of this chapter is twofold: firstly, we highlight a previously unexplored area of PU learning (augmented PU prediction) where labels for examples are available at prediction time. Secondly, we propose a novel d_B^{PU} decision rule tailored for this setting. We study the basic properties of the proposed rule and contrast it with the properties of the usual Bayes rule based solely on features of the examples. We also show that d_B^{PU}'s usefulness is not limited to the augmented PU prediction scenario, and it can be employed also in e.g. in traditional PU setting as a part of VAE-PU-Bayes model. The latter half of the chapter focuses on the practical experiments, combining d_B^{PU} rule with preexisting PU models. We start off by showing the substantial improvements of VP-B over the VAE-PU+OCC baseline for traditional PU tasks. In augmented PU prediction setting, we identify VP-B+S model as the most promising among the newly constructed methods. By comparing it with its naive counterpart, VP-B, we show that using d_B^{PU} rule systematically improves accuracy on the test dataset. However, results for the two Prophet methods (which use perfect knowledge of $y(x)$ and $s(x)$), as well as semi-Prophets (utilizing the perfect knowledge of only one of those variables) indicate that those improvements could be potentially be significantly larger, especially for the high label frequencies.

Chapter 7

Label shift for augmented PU data

7.1 Introduction

In this chapter we consider another departure from a classic Positive-Unlabled classification model which frequently occurs in practice and need to be accounted for – the distribution of data on which classifiers are trained may differ from the distribution of data to be classified, in particular a value of prior probability may change. Here, we propose modeling scenario which incorporates these nonstandard data features and show how to construct optimal classifiers in this case. We will focus on the label shift problem in the context of the augmented PU learning problem introduced in the previous chapter. We will show that a similar approach based on the d_B^{PU} decision rule with a modified threshold can be used to improve the performance of PU classifiers in the presence of label shift.

We first introduce basic notations for label shift problem. As in a standard problem, X be a random variable corresponding to feature vector, $Y \in \{-1,1\}$ be a true class label and $S \in \{0,1\}$ an indicator of an example being labeled ($S=1$) or not ($S=0$). We use the standard single-sample PU assumption, meaning that there is some unknown probability distribution P_{XYS} such that only positive examples ($Y=1$) can be labeled, i.e. $P(S=1|X,Y=-1)=0$. Moreover, we consider the second vector $(\widetilde{X},\widetilde{Y})$ such that its distribution $P_{\widetilde{X}\widetilde{Y}}$ is label-shifted distribution of P_{XY}, which means that marginal distribution of $\widetilde{Y}$ is different from that of Y i.e.

$$\widetilde{\pi} := P(\widetilde{Y}=1) \neq P(Y=1) =: \pi, \tag{7.1}$$

however we assume distributions in a positive and negative class (i.e. conditional distributions of predictors given class indicator) are the same for both distributions:

$$P_{\widetilde{X}|\widetilde{Y}=i} = P_{X|Y=i}, \quad i=\pm 1. \tag{7.2}$$

Although the proposed method is applicable also in the case $\widetilde{\pi} = \pi$, in order to streamline the presentation, we assume (7.1). Note that X and $\widetilde{X}$ correspond to the vectors of variables defined in the same way, which, have however different distributions in the image space, the same convention refers to Y and $\widetilde{Y}$. Since $P_X = \pi P_{X|Y=1} + (1-\pi)P_{X|Y=-1}$ and analogous expression holds for $P_{\widetilde{X}}$, marginal distributions of X and $\widetilde{X}$, in contrast to conditional distributions, also differ: $P_X \neq P_{\widetilde{X}}$. This can correspond e.g. to the same population characteristics in two different countries.

Such situation occurs frequently in practice. Consider e.g. the anti-causal case when X denotes symptoms of a certain disease and $Y = 1$ when it occurs. Then when prevalence of disease changes but characteristics of the symptoms of disease do not, this corresponds to label shift scenario. We assume that the second vector is not fully observable either, in the sense that, as in the case of the first vector, only positive class labels are only partially available, and denote labeling variable by $\widetilde{S}$ in this case.

Such a scenario corresponds to practical situations – for an example, let us recall the hypertension example from Chapter 6. People who check their blood pressure regularly and if its abnormal, report this to a doctor and are treated. Remaining group consists of people who are healthy and those who have hypertension but do not contact a doctor. Having such augmented data for two consecutive periods of time, we introduce a label shift factor – for example, we would like to detect those in the second group who suffer from hypertension, but allowing for a possible change of its prevalence in the population considered. Note that this scenario is applicable in many other cases like COVID-19 detection based on certain number of symptoms (coughing, difficulty of breathing) and patient's characteristics, without the necessity of performing COVID-19 test. Types of label shift in these examples might vary too, allowing for a shift in time, tested populations, or geographical regions.

As in the previous chapters, we consider a realistic no-SCAR setting where labeling mechanism may be object dependent i.e. selection bias occurs. In the following, we assume that the labeling which censors positive observations acts in the same manner in the first and in the second case, namely

$$\begin{aligned} e(x) &:= P(S=1|Y=1, X=x) \\ &= P(\widetilde{S}=1|\widetilde{Y}=1, \widetilde{X}=x) =: \widetilde{e}(x) \end{aligned} \tag{7.3}$$

i.e. that propensity scores $e(x)$ and $\widetilde{e}(x)$ are the same and they will be denoted by $e(x)$ henceforth. Note that as P_{XYS} is characterized by marginal distribution P_Y and conditional distributions $P_{X|Y}$ and $P_{S|XY}$, the probabilistic structures of P_{XYS} and $P_{\widetilde{X}\widetilde{Y}\widetilde{S}}$ are uniquely determined by the assumptions above. We also define posterior probabilities of $\widetilde{Y} = 1$ given $\widetilde{X} = x$ as $\widetilde{y}(x) = P(\widetilde{Y} = 1|\widetilde{X} = x)$ and $\widetilde{s}(x) = P(\widetilde{S} = 1|\widetilde{X} = x)$, analogously to $y(x)$ and $s(x)$ used in the standard problem. We denote by f_X either a density of X or its probability mass function and the same convention applies to $f_{X|S=1}$.

Note that it follows from Lemma 2.1 that

$$\widetilde{s}(x) = e(x)\widetilde{y}(x). \tag{7.4}$$

Assume that $\mathscr{D} = (X_i, Y_i, S_i), i = 1, \ldots, n$ is an iid sample drawn from distribution P_{XYS} and $\widetilde{\mathscr{D}} = (\widetilde{X}_i, \widetilde{Y}_i, \widetilde{S}_i), i = 1, \ldots, m$ is iid sample drawn from distribution $P_{\widetilde{X}\widetilde{Y}\widetilde{S}}$ independently of $\mathscr{D}$. Observed data consists of $(X_i, S_i), i = 1, \ldots, n$ and $(\widetilde{X}_i, \widetilde{S}_i), i = 1, \ldots, m$, thus in both cases only partial information on labels is available. Our aim is to construct a classification rule and predict class indicator $\widetilde{Y} = \pm 1$ for observations in $\widetilde{\mathscr{D}}$. Note that since $\widetilde{S}_i = 1 \Rightarrow \widetilde{Y}_i = 1$, just like in the augmented case with no shift, the task reduces to classification of unlabeled observations $(\widetilde{S}_i = 0)$ in $\widetilde{\mathscr{D}}$.

We stress that under the standard label shift probability scenario, one observes examples $(X_i, Y_i), i = 1, \ldots, n$ and $\widetilde{X}_i, i = 1, \ldots, m$ and the task is to predict class indicators for the second set for which the prior is shifted and for which class indicators are missing. For representative examples of methods designed for such scenario we refer to Saerens, Latinne, and Decaestecker (2002), Lipton, Wang, and Smola (2018) and Garg et al. (2020), see also Iyer, Nath, and Sarawagi (2014), Vaz, Izbicki, and Stern (2019), Ye et al. (2024) and references therein. We note that estimation of shifted prior probability (quantification task) is of importance in business applications (cf e.g. González et al. (2017)). We consider the label shift in the augmented PU scenario introduced in Chapter 6. To the best of our knowledge, despite its practical importance, label shift for augmented PU data has not been analyzed in the literature. The main contributions of this chapter are: (i) we construct a model for PU data which takes into account selection bias and potential label shift of target data; (ii) we establish properties of probabilistic structure of main entities in the constructed model and a form of Bayes classifier for unlabeled observations in target data; (iii) we consider empirical counterparts of the Bayes rule using different estimators of prior probability for target data and compare their behaviour on real data sets.

7.2 Main theoretical results

Below we present some basic facts concerning the label-shift augmented PU model (Section 7.2.1) and form of the Bayes classifiers under this scenario (Section 7.2.2).

7.2.1 General results

Lemma 7.1 below describes the basic facts on interplay between P_{XYS} and $P_{\widetilde{X}\widetilde{Y}\widetilde{S}}$, which will be useful for construction of classification rule based on $(\widetilde{X}, \widetilde{S})$. In particular, we prove in part (ii) that distribution $P_{\widetilde{X}\widetilde{Y}|\widetilde{S}=0}$ is label-shifted distribution of $P_{XY|S=0}$. Denote by $c = P(S = 1|Y = 1)$

and $\widetilde{c} = P(\widetilde{S} = 1|\widetilde{Y} = 1)$ overall conditional probabilities of being labeled for the first and second set, respectively. Moreover, define odds of positive class occurring for the first set as $\mathrm{OD}(x) = P(Y = 1|x)/P(Y = -1|x)$ with $\widetilde{\mathrm{OD}}(x)$ defined analogously for the second set. Odds Ratio $\mathrm{OR}(x)$ equals $\mathrm{OR}(x) = \widetilde{\mathrm{OD}}(x)/\mathrm{OD}(x)$.

Lemma 7.1. *The following equalities hold:*

(i) $\widetilde{c} = c$*,*

(ii) Assume that $\widetilde{\pi} \neq \pi$*. Then distribution* $P_{\widetilde{X}\widetilde{Y}|\widetilde{S}=0}$ *is label shifted distribution of* $P_{XY|S=0}$*. Namely, we have*

$$\begin{aligned} P(\widetilde{Y} = 1|\widetilde{S} = 0) &= \frac{\widetilde{\pi} - c\widetilde{\pi}}{1 - c\widetilde{\pi}} \\ \neq P(Y = 1|S = 0) &= \frac{\pi - c\pi}{1 - c\pi} \end{aligned}$$

and

$$\begin{aligned} f_{\widetilde{X}|\widetilde{Y}=1,\widetilde{S}=0}(x) &= f_{X|Y=1,S=0}(x), \\ f_{\widetilde{X}|\widetilde{Y}=-1,\widetilde{S}=0}(x) &= f_{X|Y=-1,S=0}(x). \end{aligned}$$

(iii) $f_{X|S=1}(x) = f_{\widetilde{X}|\widetilde{S}=1}(x)$*,*

(iv) $OD(x)\frac{1-\pi}{\pi} = \widetilde{OD}(x)\frac{1-\widetilde{\pi}}{\widetilde{\pi}} \equiv OR(x) = \frac{\widetilde{\pi}}{1-\widetilde{\pi}} \times \frac{1-\pi}{\pi}$*.*

Property (iv) is a standard result for label shift, see e.g. Elkan (2001). For completeness, we give its proof below.

Proof. (i) Note that

$$\begin{aligned} c &= \int P(S = 1|Y = 1, X = x) f_{X|Y=1}(x)\,dx \\ &= \mathbb{E}_{X|Y=1}\, e(X) = \mathbb{E}_{\widetilde{X}|\widetilde{Y}=1}\, e(\widetilde{X}) = \widetilde{c}, \end{aligned}$$

where the first equality follows from definitions of c and $e(x)$, the second from assumed equality of distributions within classes, and the last one from equality $\widetilde{e}(x) = e(x)$.

(ii) To prove the first part, note that $P(\widetilde{S} = 0) = 1 - c\widetilde{\pi}$, which yields expressions for conditional probabilities. Moreover, inequality of the conditional probabilities follows from strict monotonicity of the function $f(a) = a/(1 - ca)$ for $a \in (0, 1)$.

For the second part, note that

$$\begin{aligned}
&f_{X|Y=1,S=0}(x)\\
&=\frac{P(Y=1,X=x)P(S=0|Y=1,X=x)}{P(Y=1,S=0)}\\
&=\frac{f_{X|Y=1}(x)\pi(1-e(x))}{P(Y=1,S=0)}=\frac{f_{X|Y=1}(x)\pi(1-e(x))}{\pi-\pi c}\\
&=f_{X|Y=1}(x)(1-e(x))(1-c)^{-1}.
\end{aligned}$$

As analogous formula holds for $f_{\widetilde{X}|\widetilde{Y}=1,\widetilde{S}=0}(x)$, in view of (7.3), $f_{X|Y=1}=f_{\widetilde{X}|\widetilde{Y}=1}$ and $c=\widetilde{c}$, the first part is proved. The second is even more straightforward as

$$\begin{aligned}
f_{X|Y=-1,S=0}(x)&=f_{X|Y=-1}(x)\\
&=f_{\widetilde{X}|\widetilde{Y}=-1}(x)=f_{\widetilde{X}|\widetilde{Y}=-1,\widetilde{S}=0}(x).
\end{aligned}$$

(iii) Note that in view of Lemma 2.1 we have

$$\begin{aligned}
f_{X|S=1}(x)&=\frac{s(x)f_X(x)}{P(S=1)}=\frac{y(x)e(x)f_X(x)}{P(S=1)}\\
&=\frac{P(Y=1,X=x)}{f_X(x)}\times\frac{e(x)f_X(x)}{P(S=1)}\\
&=f_{X|Y=1}(x)\frac{\pi e(x)}{P(S=1)}.
\end{aligned}$$

Replacing in the last expression $f_{X|Y=1}(x)$ by $f_{\widetilde{X}|\widetilde{Y}=1}(x)$ and repeating the above line of argument backwards we obtain

$$\begin{aligned}
f_{X|S=1}(x)&=f_{\widetilde{X}|\widetilde{S}=1}(x)\frac{\pi}{P(S=1)}\frac{P(\widetilde{S}=1)}{\widetilde{\pi}}\\
&=f_{\widetilde{X}|\widetilde{S}=1}(x)\times\widetilde{c}/c=f_{\widetilde{X}|\widetilde{S}=1}(x),
\end{aligned}$$

where the last equality follows from (i).

(iv) Reasoning as before, we note that

$$\widetilde{y}(x)=\frac{f_{\widetilde{X}|\widetilde{Y}=1}(x)\widetilde{\pi}}{f_{\widetilde{X}}(x)}=\frac{y(x)f_X(x)\widetilde{\pi}/\pi}{f_{\widetilde{X}}(x)}$$

and

$$1-\widetilde{y}(x)=\frac{(1-y(x))f_X(x)(1-\widetilde{\pi})/(1-\pi)}{f_{\widetilde{X}}(x)}.$$

Dividing the expression above yields the conclusion.

□

Remark 7.2. *(i) We note that important equality in Lemma 7.1.(iii) can be intuitively justified by noting that labeled ($S = 1$) observation $X = x$ is picked from the strata $Y = 1$ described by distribution $f_{X|Y=1}$ with probability $e(x)$. As in the case of label-shifted distribution, distributions of positive class and labeling mechanism are the same as for P_{XYS}, the conclusion follows.*

(ii) Note that Lemma 7.1.(i) implies, as we have $c = P(S = 1)/P(Y = 1)$, that if $\widetilde{\pi} > \pi$ then $P(\widetilde{S} = 1) > P(S = 1)$ and vice versa. Moreover, proportion of positives to negatives equals $\pi(1-c)/(1-\pi)$ for unlabeled population $S = 0$ and $\widetilde{\pi}(1-c)/(1-\widetilde{\pi})$ for $\widetilde{S} = 0$ (see Lemma 7.1.(ii)).

(iii) The ratio $\frac{\widetilde{\pi}}{1-\widetilde{\pi}} \times \frac{1-\pi}{\pi}$ appearing in Lemma 7.1.(iv) can be interpreted in term of dataset imbalance. Namely, note that a ratio of the form $I(\mathscr{D}) = \frac{\pi}{1-\pi}$ is a measure of imbalance of the dataset – values close to 1 indicate a balanced dataset, whereas very big (small) values indicate a large proportion of positive (negative) examples, respectively. Thus, the ratio $\frac{\widetilde{\pi}}{1-\widetilde{\pi}} \times \frac{1-\pi}{\pi}$ can be interpreted as an imbalance ratio IR between the two datasets: $IR = \frac{I(\widetilde{\mathscr{D}})}{I(\mathscr{D})}$.

We also note that the stronger property than Lemma 7.1.(i) holds, namely for any Borel set A, $P(S = 1|Y = 1, X \in A) = P(\widetilde{S} = 1|\widetilde{Y} = 1, \widetilde{X} \in A)$. Moreover, note that no label-shift situation $\widetilde{\pi} = \pi$ is equivalent in the view of Lemma 7.1.(i) to $P(S = 1) = P(\widetilde{S} = 1)$ which can be routinely tested using difference of two binomial proportion test (this does not require knowledge of prior π).

In the following, as before, we assume that the prior probability $\pi = P(Y = 1)$ is known. This is reasonable assumption when $Y = 1$ corresponds to disease and its prevalence can be estimated with arbitrary accuracy. We note in passing that in this case distribution of negative examples $P_{X|Y=0} = (1-\pi)^{-1}\left(P_X - \pi P_{X|Y=1}\right)$ is identifiable although no set pertaining to it is available. Denote $\gamma = P(S = 1)$ and $\widetilde{\gamma} = P(\widetilde{S} = 1)$. Then Lemma 7.1.(i) can be rewritten as

$$\widetilde{\pi} = \frac{P(\widetilde{S} = 1)}{P(S = 1)} \times \pi, \tag{7.5}$$

thus yielding plug-in estimator of $\widetilde{\pi}$:

$$\begin{aligned} \widehat{\widetilde{\pi}} &= \frac{\widehat{\widetilde{\gamma}}}{\widehat{\gamma}} \mathbb{I}\{\widehat{\gamma} > 0\} \times \pi \\ &= \frac{\#(i : \widetilde{S}_i = 1)/m}{\#(i : S_i = 1)/n} \mathbb{I}\{\widehat{\gamma} > 0\} \times \pi. \end{aligned} \tag{7.6}$$

The Lemma 7.3 below lists the basic properties of $\widehat{\widetilde{\pi}}$. For a proof, refer to the full paper (Mielniczuk and Wawrzeńczyk 2025a).

Lemma 7.3. *(i) We have for any $\delta > 0$ that with probability at least $1-\delta$*

$$|\widehat{\widetilde{\pi}} - \widetilde{\pi}| \leq \tfrac{1}{c}\left(\tfrac{1}{\widehat{\gamma}}\sqrt{\tfrac{1}{n}\log\left(\tfrac{4}{\delta}\right)} + \sqrt{\tfrac{1}{m}\log\left(\tfrac{4}{\delta}\right)}\right) \tag{7.7}$$

and thus the rate of almost sure convergence of $\widehat{\widetilde{\pi}}$ to $\widetilde{\pi}$ is $\min(n,m)^{-1/2}$.

(ii) We have, with $\widetilde{\gamma} = P(\widetilde{S}=1)$

$$\mathbb{E}\,\widehat{\widetilde{\pi}} = \widetilde{\gamma}\,\mathbb{E}\left(\widehat{\gamma}^{-1}\mathbb{I}\{\widehat{\gamma} > 0\}\right) \times \pi = \widetilde{\pi}\left(1 + \mathscr{O}\left(\frac{1}{n}\right)\right).$$

7.2.2 Bayes classification rules for label-shift augmented PU data

Let $\eta(x) = P(Y=1|S=0, X=x)$ and $\widetilde{\eta}(x) = P(\widetilde{Y}=1|\widetilde{S}=0, \widetilde{X}=x)$. In Chapter 6, Bayes classification function $d_B^{PU}(x) = \eta(x)/(1-\eta(x))$ based on (X,S) was considered on strata $S=0$. It follows that the corresponding Bayes rule has the following form: the observation is classified to the positive class, if the condition (see (6.8))

$$d_B^{PU}(x) = \frac{\eta(x)}{1-\eta(x)} = \frac{y(x)-s(x)}{1-y(x)} > 1 \tag{7.8}$$

is satisfied, and to the negative class in the opposite case. Here we show that in the label shift case the rule is modified by changing the threshold of the $d_B^{PU}(x)$.

Theorem 7.4. *(i) The Bayes rule for $(\widetilde{X},\widetilde{S}) = (x,0)$ has the following form. Classify $(x,0)$ to class $Y=1$ if the condition*

$$\begin{aligned} d_B^{PU}(x) &> \frac{\pi}{1-\pi}\,\frac{1-\widetilde{\pi}}{\widetilde{\pi}} \\ &= \frac{\frac{P(S=1)}{P(\widetilde{S}=1)} - \pi}{1-\pi} = \frac{\pi - \pi\widetilde{\pi}}{\widetilde{\pi} - \pi\widetilde{\pi}} =: \theta \end{aligned} \tag{7.9}$$

is satisfied, or formulating the rule equivalently, $y(x) > (\theta + s(x))/(1+\theta) = \frac{\pi - \pi\widetilde{\pi}}{\pi + \widetilde{\pi} - 2\pi\widetilde{\pi}}$.

(ii) The Bayes risk of $\widetilde{d}_B^{PU}$ equals

$$\begin{aligned} &P(\widetilde{S}=0)\,\mathbb{E}_{\widetilde{X}|\widetilde{S}=0}\min\left(\widetilde{\eta}(\widetilde{X}), 1-\widetilde{\eta}(\widetilde{X})\right) \\ &= \frac{1}{2}P(\widetilde{S}=0) - \frac{1}{2}\mathbb{E}_{\widetilde{X}\widetilde{S}}\left|2\widetilde{\eta}(\widetilde{X}) - 1\right|. \end{aligned}$$

Note that if no label shift occurs, then in Eq. (7.9) the threshold $\theta = 1$, thus the result generalizes Theorem 6.1.(ii) in Chapter 6.

Proof. (i) Let $\widetilde{d}_B^{PU}(x) = \widetilde{\eta}(x)/(1-\widetilde{\eta}(x))$ be a Bayes classification function corresponding to $\widetilde{\eta}(x)$. The pertaining Bayes rule has the following form

$$\begin{aligned} 1 < \frac{\widetilde{\eta}(x)}{1-\widetilde{\eta}(x)} &= \frac{P(\widetilde{Y}=1|\widetilde{S}=0,\widetilde{X}=x)}{P(\widetilde{Y}=-1|\widetilde{S}=0,\widetilde{X}=x)} \\ &= \frac{P(\widetilde{Y}=1,\widetilde{S}=0,\widetilde{X}=x)}{P(\widetilde{Y}=-1,\widetilde{S}=0,\widetilde{X}=x)} \\ &= \frac{f_{\widetilde{X}}(x)}{f_{\widetilde{X}}(x)} \times \frac{\widetilde{y}(x)-\widetilde{s}(x)}{1-\widetilde{y}(x)} = \frac{\widetilde{y}(x)(1-e(x))}{1-\widetilde{y}(x)} \\ &= \frac{y(x)(1-e(x))}{1-y(x)} \frac{1-\pi}{\pi} \frac{\widetilde{\pi}}{1-\widetilde{\pi}}, \end{aligned}$$

where the last equality follows from Lemma 7.1.(iv). This is, using Lemma 7.1.(i), equivalent to the following event

$$\begin{aligned} \frac{y(x)-s(x)}{1-y(x)} &> \frac{\pi}{\widetilde{\pi}} \frac{1-\widetilde{\pi}}{1-\pi} \\ &= \frac{P(S=1)}{P(\widetilde{S}=1)} \frac{(1-\pi P(\widetilde{S}=1)/P(S=1))}{1-\pi} \\ &= \frac{P(S=1)-\pi P(\widetilde{S}=1)}{P(\widetilde{S}=1)(1-\pi)} = \frac{\frac{P(S=1)}{P(\widetilde{S}=1)}-\pi}{1-\pi}. \end{aligned}$$

Alternatively, using Lemma 7.1.(iv) we have:

$$\widetilde{\mathrm{OD}}(x) = \mathrm{OD}(x)\frac{1-\pi}{\pi}\frac{\widetilde{\pi}}{1-\widetilde{\pi}} =: \mathrm{OD}(x)\psi$$

Using this, we can reason as follows to obtain the threshold:

$$\begin{gathered} \widetilde{\mathrm{OD}}(x) > 1 \equiv \mathrm{OD}(x)\,\psi > 1 \\ \psi\frac{y(x)}{1-y(x)} > 1 \\ \psi y(x) > 1 - y(x) \\ y(x) > \frac{1}{1+\psi} = \frac{1}{1+\frac{1-\pi}{\pi}\frac{\widetilde{\pi}}{1-\widetilde{\pi}}} = \frac{1}{1+\frac{\widetilde{\pi}-\pi\widetilde{\pi}}{\pi-\pi\widetilde{\pi}}} = \frac{1}{\frac{\pi-\pi\widetilde{\pi}+\widetilde{\pi}-\pi\widetilde{\pi}}{\pi-\pi\widetilde{\pi}}} = \frac{\pi-\pi\widetilde{\pi}}{\pi+\widetilde{\pi}-2\pi\widetilde{\pi}}. \end{gathered}$$

(ii) Proof follows from Theorem 6.1 in Chapter 6.

□

Theorem 7.4.(i) can be explained in the following way. As $P_{\widetilde{X}|\widetilde{S}=0}$ is label shifted distribution of $P_{X|S=0}$ and decisions on $S=1$ and and $\widetilde{S}=1$ are error-free, the Bayes rule for the target population is Bayes rule for the source with changed threshold. The Bayes rule in question is $d_B^{PU}(x)$, and the modified threshold for label shifted population is in this case (see e.g. Elkan (2001)):

$$\frac{P(Y=1|S=0)}{P(Y=-1|S=0)}\frac{P(\widetilde{Y}=-1|\widetilde{S}=0)}{P(\widetilde{Y}=1|\widetilde{S}=0)}=\frac{\pi}{1-\pi}\frac{1-\widetilde{\pi}}{\widetilde{\pi}},$$

where the last equality follows from Lemma 7.1.(i). This coincides with (7.9). Note that, surprisingly at the first sight, the threshold does not depend on c. This is due to the fact that the imbalance ratio for the target and training set equal $(\widetilde{\pi}/(1-\widetilde{\pi}))/(\pi/(1-\pi))$ is the same as imbalance ratio for their corresponding unlabeled subsets (see Remark 7.2 (ii)).

Note that if $\widetilde{\pi}>\pi$ then the rule becomes less conservative than in the no-label-shift case. Moreover, it follows from the proof above that the Bayes rule which (erroneously) does not take into account the label shift will classify to positive class if

$$\frac{y(x)-s(x)}{1-y(x)}>1 \quad \equiv \quad \frac{\widetilde{y}(x)-\widetilde{s}(x)}{1-\widetilde{y}(x)}>\frac{\widetilde{\pi}}{1-\widetilde{\pi}}\frac{1-\pi}{\pi}.$$

Rule in (7.9) yields its empirical analogue for label-shift case: classify as positive ($Y=1$) when

$$\frac{\widehat{y}(x)-\widehat{s}(x)}{1-\widehat{y}(x)}>\frac{\frac{N_1}{n}\frac{m}{M_1}-\pi}{1-\pi}, \tag{7.10}$$

where N_1, M_1 are sizes of labeled sets in $\mathscr{D}$ and $\widetilde{\mathscr{D}}$, respectively and $\widehat{y}(x)$, $\widehat{s}(x)$ are estimators of $y(x)$ and $s(x)$ discussed below.

7.3 Experiments

7.3.1 Datasets

To estimate the performance of the label shift methods, we based our experiments on several datasets with varying characteristics, described in detail in Appendix E (we keep the naming convention from the previous chapter). We used several settings for the label frequency: $c\in\{0.1,0.3,0.5,0.7,0.9\}$.

We simulate label shift phenomenon as follows. In our experiments, we assume several $\widetilde{\pi}$ values ($\widetilde{\pi}\in\{0.1,0.3,0.5,0.7,0.9\}$, as well as the original dataset, corresponding to $\widetilde{\pi}=\pi$). At test time we obtain label shifted dataset:

- $\boldsymbol{\widetilde{\pi}>\pi}$. In order to increase the prior in the test dataset, we drop a portion of negative examples. We randomly sample $\frac{1-\widetilde{\pi}}{\widetilde{\pi}}\times\#(Y_{\text{test}}=1)<\#(Y_{\text{test}}=-1)$ negative examples. As the ratio of positives to negatives is now approximately $n\pi/[n\pi(1-\tilde{\pi})/\tilde{\pi})]$, we obtain a label shifted example with prior $\tilde{\pi}$.

- $\tilde{\boldsymbol{\pi}} < \boldsymbol{\pi}$. Symmetric case. We decrease the proportion of positive examples to $\tilde{\pi}$ by sampling $\frac{\tilde{\pi}}{1-\tilde{\pi}} \times \#(Y_{\text{test}} = -1) < \#(Y_{\text{test}} = 1)$ positives.

- $\tilde{\boldsymbol{\pi}} = \boldsymbol{\pi}$. We preserve the original test dataset.

7.3.2 Baseline models

VAE-PU-Bayes (abbreviated to VP-B) introduced in Chapter 6 was selected as a state-of-the-art PU method working under SAR assumption. Estimation of $s(x)$ is performed using a simple feed forward Neural Network (NN).

7.3.3 $\tilde{\pi}$ estimation

We start by considering estimators of $\tilde{\pi}$ which will be then incorporated into proposed classifiers. The first estimator is $\widehat{\tilde{\pi}}$ defined in (7.6) and is called the Direct estimator of $\tilde{\pi}$ in the following. Alternatively, one can apply classical EM algorithm, cf Saerens, Latinne, and Decaestecker (2002), to estimate labeled-shifted prior probability $\tilde{\pi}$ in this setting. For this aim $y(x)$ (which in classical case is estimated by e.g. NN based on fully observable set $(X_i, Y_i), i = 1, \ldots, n)$ is estimated here by one of the PU estimators of posterior probability under selection bias. For this purpose, we use Variational PU-Bayes classifier VP-B (see Section 6.3), but other variational classifiers designed for this framework (see Na et al. (2020) and Wawrzeńczyk and Mielniczuk (2023a)) can be used as well. Note that as π it is assumed known, it replaces in the original algorithm fraction of positive observation in the dataset, which is not available. The resulting EM estimator is denoted by $\widehat{\tilde{\pi}}_{EM}$. We note that due to the use of the EM algorithm this approach is more computationally expensive than the Direct estimator.

7.3.4 Empirical Bayes rules classifiers

We define several classifiers based derivations in Section 7.2.1. In our experiments, we aimed to evaluate the proposed methods and choose the one most adequate to handle the biased label shift PU problems.

CLS estimator This is an empirical analogue of Bayes rule defined in (7.10) with $\widehat{\tilde{\pi}}$ defined in (7.6). We use VP-B to estimate $y(x)$ and a separate NN to estimate $s(x)$. The proposal is named Cut-off Label Shift (CLS) estimator.

CLS-EM estimator The estimator is defined similarly to CLS, the only difference being that threshold is changed from $\widehat{\widetilde{\pi}}$ to $\widehat{\widetilde{\pi}}_{EM}$.

ALS estimator We note that due to Lemma 2.1 and (7.4) we have

$$\widetilde{y}(x) = \frac{\widetilde{s}(x)}{s(x)} \times y(x). \tag{7.11}$$

Plugging-in this expression in expression for the classification function $\widetilde{d}_B^{PU}(x)$ we obtain (cf (7.8))

$$\frac{\widetilde{y}(x) - \widetilde{s}(x)}{1 - \widetilde{y}(x)} > 1 \quad \equiv \quad \frac{\widetilde{s}(x)\left(\frac{y(x)}{s(x)} - 1\right)}{1 - \frac{\widetilde{s}(x)}{s(x)} y(x)} > 1$$

This gives rise to the competing empirical Bayes rules: estimate $y(x)$ and $s(x)$ based on $\mathscr{D}$ and $\widetilde{s}$ based on $\widetilde{D}$ and apply the formula above with plugged in estimators to construct empirical Bayes rule. We use VP-B to estimate $y(x)$ and separate NNs to estimate $s(x)$ and $\widetilde{s}(x)$. We note that in contrast to the classifiers introduced above one needs to estimate posterior $\widetilde{s}(x)$. We call this estimator Augmented Label Shift (ALS) estimator.

7.3.5 General experiment settings

For each experimental setting (i.e. a combination of dataset, label frequency c, target label shift prior $\widetilde{\pi}$ and label shift estimator), we performed 10 experiments, each initialized with a different random seed (equal to experiment number). Data was split between train and test following 70-30 split. Because prediction for labeled examples is trivial in this setting (as $S = 1$ implies $Y = 1$), instead of using traditional metrics, as in Chapter 6, we consider a set of U-metrics, calculated based only on unlabeled stratum of test set, which eliminates trivial prediction impact and puts focus on the classifier performance on the key test subset. All method and experiment code is available in a public GitHub repository[1].

7.4 Results of experiments

Table 7.1 and Figure 7.1 contrast the $\widetilde{\pi}$ estimation performance of the Direct and EM estimators. Both achieve generally good results, and in both cases, their quality increases as label frequency rises. This is especially apparent for Direct estimator and due to the fact that for low c values the number of labeled examples is low, and thus estimation of $P(S = 1)$ in the denominator of (7.5) becomes sensitive to small deviations. Also, this is consistent with form of the bound in (7.7),

[1] https://github.com/wawrzenczyka/VAE-PU-label-shift

Table 7.1: MSE of Direct and EM estimators of $\widetilde{\pi}$ (mean over all datasets)

c	0.1		0.3		0.5		0.7		0.9	
Estimator	Direct	EM	Direct	EM	Direct	EM	Direct	EM	Direct	EM
$\widetilde{\pi}$										
0.1	**0.024**	0.195	**0.013**	0.159	**0.007**	0.150	**0.006**	0.140	**0.003**	0.130
0.3	**0.035**	0.121	**0.023**	0.092	**0.013**	0.088	**0.010**	0.081	**0.005**	0.075
0.5	**0.039**	0.054	**0.025**	0.029	**0.015**	0.029	**0.012**	0.025	**0.005**	0.022
0.7	**0.052**	0.053	**0.028**	0.049	**0.018**	0.042	**0.014**	0.040	**0.006**	0.040
0.9	**0.066**	0.117	**0.036**	0.114	**0.024**	0.103	**0.017**	0.097	**0.007**	0.095
No shift	**0.034**	0.054	**0.019**	0.029	**0.012**	0.028	**0.009**	0.025	**0.004**	0.021
Mean error	**0.042**	0.099	**0.024**	0.079	**0.015**	0.073	**0.011**	0.068	**0.005**	0.064

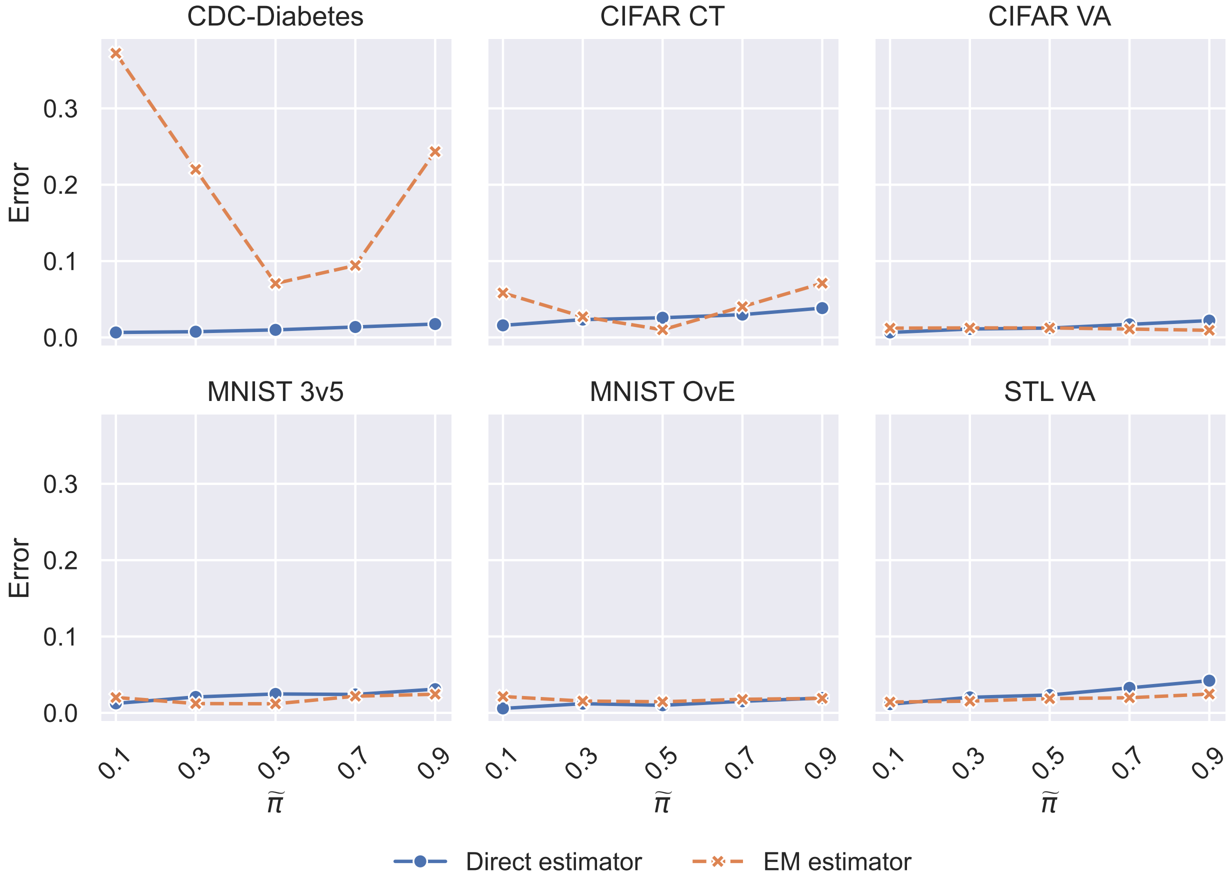


Figure 7.1: $\widetilde{\pi}$ estimation errors (MSE) comparison (averaged over all label frequencies)

Table 7.2: U-Balanced accuracy ranks

Estimator	CDC-Diabetes	CIFAR CT	CIFAR VA	MNIST 3v5	MNIST OvE	STL VA	Mean Rank
ALS	2.29	2.64	2.74	2.72	2.73	2.76	2.65
CLS	**1.68**	**1.61**	1.66	**1.48**	**1.37**	**1.49**	**1.55**
CLS-EM	2.03	1.75	**1.60**	1.80	1.89	1.75	1.80

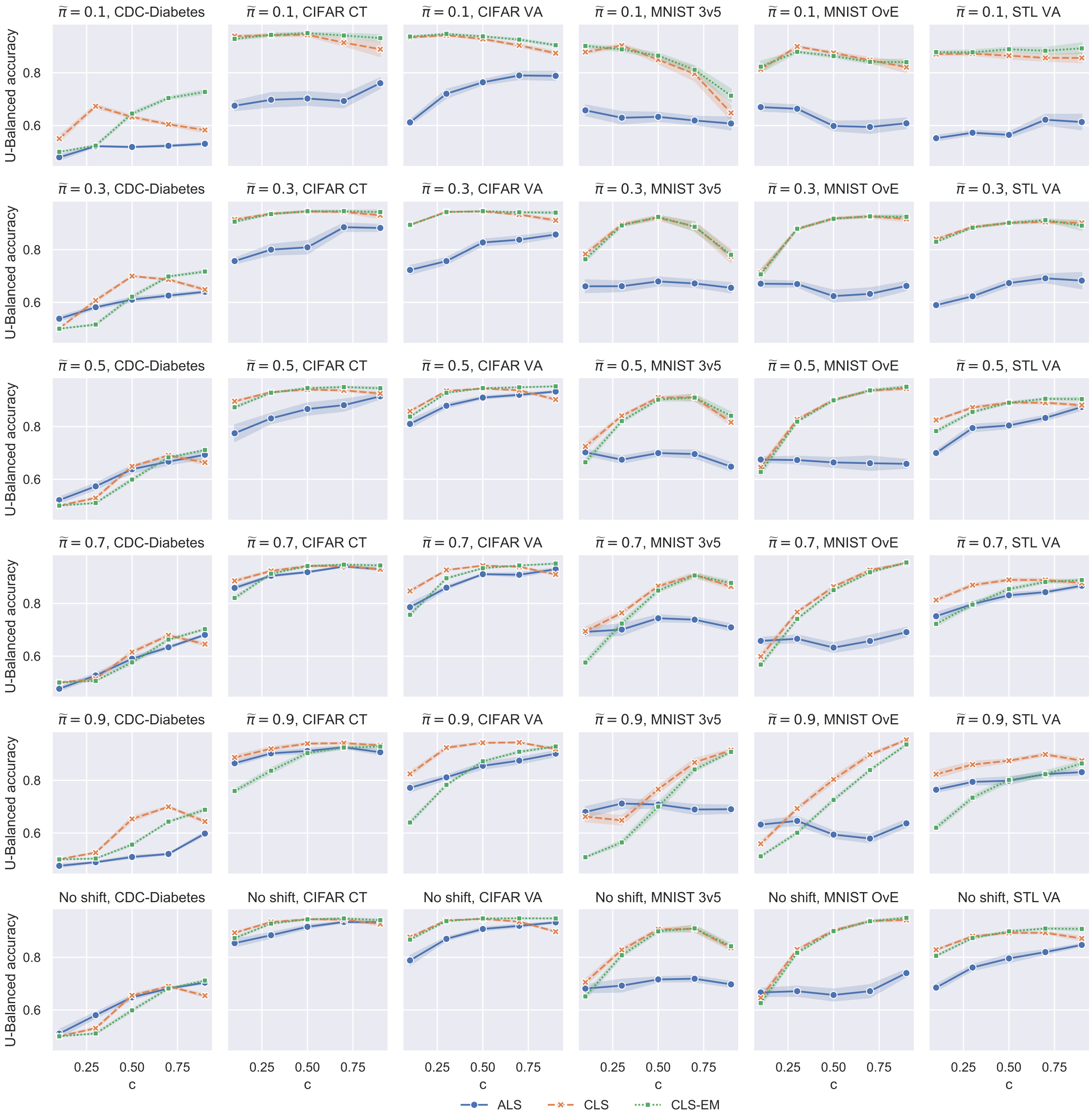


Figure 7.2: U-Balanced accuracy in label shift scenario for each estimator

Table 7.3: U-Balanced accuracy difference from best estimator

Estimator	CDC-Diabetes	CIFAR CT	CIFAR VA	MNIST 3v5	MNIST OvE	STL VA	Mean Difference
ALS	0.059	0.086	0.089	0.156	0.197	0.142	0.122
CLS	**0.027**	**0.010**	**0.011**	**0.015**	**0.010**	**0.010**	**0.014**
CLS-EM	0.034	0.018	0.021	0.033	0.024	0.030	0.027

where c appears in the denominator of the bound. As evident in Figure 7.1, with the exception of CDC-Diabetes dataset, the Direct estimator also tends to perform better for lower shift priors, with EM outperforming Direct estimator for the higher prior values. Both this property and the significant variability of results are heavily impacted by the low label frequency performance of both estimators: compare MSEs of both estimators for $c = 0.1$ and $c = 0.9$ in Table 7.1. An important thing to note is that using the EM estimator is associated with the additional cost of the iterative EM procedure – it increases the computational complexity of the estimation, making it significantly slower compared to the Direct method.

Figure 7.2 compares the performance of the three estimators proposed in Section 7.3.4. It is evident that the ALS estimator lags behind the other two methods in terms of U-Balanced accuracy. This is likely to be related to its more complex formulation, requiring training of two additional models for the estimation of both $s(x)$ and $\widetilde{s}(x)$. CLS and CLS-EM are significantly more comparable in performance. To help differentiate between them, we report average rank in Table 7.2, as well as the difference of accuracy between the method and the best classifier in Table 7.3. Both metrics are averaged over all label frequencies and label shift priors. According to both metrics, CLS outperforms the competitor, with a mean rank of around 1.5 and approximately two times smaller difference from the best estimator. CLS estimator is more consistent over the performed experiments and various datasets, but CLS-EM also has evident benefits in some cases - it often slightly edges out the CLS estimator for high label frequencies. We note also consistently better performance of CLS for $\widetilde{\pi} = 0.9$ (the fifth row of Figure 7.2). For the case of no shift ($\widetilde{\pi} = \pi$, the last row of Figure 7.2) their performance is strikingly similar, with slight superiority of CLS-EM for $c \geq 0.7$ in the case of CIFAR VA and STL VA.

Table 7.4 shows the maximum standard error of the mean U-Balanced accuracy for each estimator, when the maximum is taken over c. The reported errors are small when compared to U-balanced accuracy, which indicates that the results are stable and reliable. The highest standard error is observed for the ALS estimator, which matches the results presented in Figure 7.2 – the CLS and CLS-EM estimators show more consistent performance.

Using MNIST 3v5 as an example, we analyzed how different labeling biases affect the performance of the estimators. While creating the dataset as described in the Section 7.3.1, instead of labeling the most bold examples, we sampled the examples without replacement with step weights. The "standard" labeling reported before corresponds to 0–1 step weight: the boldest n_L examples are sampled with weight 1, while the remaining have their weights set to 0. Similarly, 0.5–0.5 step corresponds to the SCAR scenario, as both the top n_L examples and the remaining ones are sampled with equal weight of 0.5. The results are presented in Table 7.5. We observe that the performance of the estimators is consistent across different labeling biases, with performance generally improving as the datasets get closer to the SCAR scenario (except for the

Table 7.4: Maximum standard error of the mean U-Balanced accuracy for each estimator

Estimator	Dataset	No shift	$\widetilde{\pi} = 0.1$	$\widetilde{\pi} = 0.3$	$\widetilde{\pi} = 0.5$	$\widetilde{\pi} = 0.7$	$\widetilde{\pi} = 0.9$
ALS	CDC-Diabetes	0.011	0.009	0.014	0.010	0.006	0.016
	CIFAR CT	0.027	0.025	0.032	0.008	0.010	0.014
	CIFAR VA	0.017	0.017	0.016	0.016	0.016	0.019
	MNIST 3v5	0.025	0.024	0.016	0.022	0.020	0.024
	MNIST OvE	0.024	0.023	0.027	0.022	0.017	0.025
	STL VA	0.029	0.031	0.014	0.014	0.015	0.016
CLS	CDC-Diabetes	0.011	0.007	0.009	0.008	0.010	0.006
	CIFAR CT	0.028	0.013	0.006	0.005	0.008	0.007
	CIFAR VA	0.007	0.006	0.005	0.006	0.009	0.003
	MNIST 3v5	0.027	0.025	0.012	0.016	0.019	0.013
	MNIST OvE	0.021	0.013	0.012	0.012	0.013	0.012
	STL VA	0.018	0.011	0.006	0.009	0.014	0.009
CLS-EM	CDC-Diabetes	0.008	0.007	0.007	0.006	0.006	0.007
	CIFAR CT	0.013	0.008	0.006	0.007	0.011	0.005
	CIFAR VA	0.004	0.005	0.003	0.005	0.007	0.003
	MNIST 3v5	0.025	0.017	0.015	0.013	0.017	0.012
	MNIST OvE	0.021	0.016	0.013	0.009	0.006	0.013
	STL VA	0.021	0.018	0.007	0.009	0.013	0.007

Table 7.5: Mean U-Balanced accuracy for MNIST 3v5 with various step probabilities and shift priors, averaged by label frequency

Estimator	Step size	No shift	$\widetilde{\pi} = 0.1$	$\widetilde{\pi} = 0.3$	$\widetilde{\pi} = 0.5$	$\widetilde{\pi} = 0.7$	$\widetilde{\pi} = 0.9$
ALS	0 – 1 (standard)	0.701	0.629	0.666	0.684	0.717	0.696
	0.1 – 0.9	0.717	0.576	0.585	0.568	0.739	0.716
	0.3 – 0.7	0.747	0.583	0.576	0.555	0.781	0.745
	0.5 – 0.5 (SCAR)	0.760	0.577	0.587	0.551	0.775	0.745
CLS	0 – 1 (standard)	0.837	0.815	0.853	0.841	0.819	0.772
	0.1 – 0.9	0.889	0.867	0.898	0.885	0.878	0.855
	0.3 – 0.7	0.913	0.908	0.920	0.909	0.902	0.890
	0.5 – 0.5 (SCAR)	0.919	0.918	0.930	0.916	0.914	0.909
CLS-EM	0 – 1 (standard)	0.822	0.836	0.850	0.828	0.787	0.704
	0.1 – 0.9	0.865	0.894	0.906	0.872	0.817	0.720
	0.3 – 0.7	0.890	0.933	0.926	0.896	0.850	0.759
	0.5 – 0.5 (SCAR)	0.900	0.940	0.931	0.905	0.865	0.784

select cases of the ALS estimator), which is obviously the easiest task to solve. This illustrates the robustness of the estimators to different labeling biases regardless of the shift prior.

7.5 Summary

In this chapter, we discuss the issue of label shift phenomenon in the context of augmented PU data. In the general result section, we investigated probabilistic structure of label-shifted augmented PU data, proving that the label shift for general populations carries over to their unlabeled subpopulations. Moreover, we constructed the correct Bayesian rule for the label-shifted set showing that it is Bayes classifier for the augmented PU data with appropriately modified threshold. Our findings led us to propose three potential classifiers built upon the state-of-the-art VAE-PU-Bayes method: ALS, CLS and CLS-EM. As an intermediate step, we also consider the problem of $\widetilde{\pi}$ estimation, which is a key component of the proposed classifiers; we propose the Direct and EM estimators in order to solve this problem. In the experiment section, we show that both the Direct and EM estimators perform well in terms of $\widetilde{\pi}$ estimation. We also conclude that the CLS estimator generally outperforms the competing methods, and CLS-EM is a viable alternative in high label frequency scenarios.

Chapter 8

Summary

8.1 Research contributions

The thesis addresses fundamental challenges in Positive-Unlabeled (PU) learning, where only some positive examples are labeled while all negative examples remain unlabeled. The research advances PU learning methods in the complex no-SCAR scenario, where labeling depends on features, making classification significantly more challenging than under traditional SCAR assumptions.

Scenario compatibility and foundational issues. The first contribution addresses a critical gap in PU learning methodology by formally analyzing the compatibility between single sample (SS) and case control (CC) scenarios. Through Proposition 3.1, the thesis proves that these scenarios are fundamentally incompatible – cross-scenario application of methods is only valid under highly restrictive conditions that never hold in practice. This theoretical result explains widespread confusion in the literature where researchers mistakenly apply methods across scenarios. The practical contribution includes developing scenario-aware variants such as nnPU_{SS}, requiring minimal code changes but providing substantial performance improvements up to 20 percentage points while dramatically reducing overfitting. This work establishes proper protocols for method development and evaluation in PU learning in the context of varying scenarios.

Generative approaches to no-SCAR PU learning. The thesis advances no-SCAR PU learning through variational autoencoders, introducing VAE-PU+OCC as a significant improvement over the baseline VAE-PU method. The key innovation replaces VAE-PU's problematic use of generated examples directly in risk estimation with a one-class classification approach that identifies true positive-unlabeled examples within the original unlabeled dataset. This eliminates the need for risk truncation while ensuring theoretical non-negativity conditions hold naturally. Through comprehensive experiments across diverse datasets, VAE-PU+OCC consistently outperforms competing methods, including SAR-EM and LBE, and achieving improvements up to 20 percentage points while providing improved precision-recall balance and reduced training times. The approach demonstrates how generative models can effectively handle the fundamental distributional mismatch in no-SCAR scenarios.

False Omission Rate control in outlier detection. The thesis introduces the first empirical procedure for False Omission Rate (FOR) control in outlier detection scenarios, addressing applications where controlling the proportion of undetected outliers among observations classified as inliers is critical. The theoretical contribution establishes the mathematical foundation for FOR control through theorems proving that FOR and Bayesian FOR are approximately equivalent for large datasets, with guaranteed existence and uniqueness of optimal thresholds under specified conditions. The practical innovation develops an empirical rule analogous to the Benjamini-Hochberg procedure, but designed for FOR rather than FDR control. Experiments across 30 diverse datasets show the method successfully controls FOR when high-quality outlier scores are available, with effectiveness roughly correlated to the skewness difference between outlier and inlier p-value distributions. Integration into the VAE-PU framework as VAE-PU+FOR demonstrates the method's broader applicability, though it reveals sensitivity to outlier score quality.

Augmented PU learning framework. The thesis identifies and formalizes augmented PU learning, where labeling status is available at prediction time – a previously unexplored but practically important scenario. The theoretical contribution develops the optimal Bayesian decision rule d_B^{PU} requiring $y(x) > \frac{1+s(x)}{2}$ for positive classification on unlabeled examples, proving this rule is more conservative than standard Bayes classification. The practical contribution, VAE-PU-Bayes, represents the thesis's most significant methodological advancement, applying the d_B^{PU} rule within the VAE-PU framework and achieving superior performance after identifying an augmented subproblem in standard PU task. Comprehensive experiments show VAE-PU-Bayes consistently outperforms existing VAE-PU variants by up to 5 percentage points in traditional PU settings, while achieving systematic improvements in augmented scenarios. This establishes a new PU learning paradigm with mathematical foundations for understanding when labeling information should influence classification decisions.

Label shift in augmented PU data. The thesis addresses label shift in augmented PU learning, where class prior probability differs between training and test datasets while maintaining identical conditional distributions within classes. The theoretical contribution establishes through mathematical analysis that label shift in general populations carries over to their unlabeled subpopulations, and demonstrates that the optimal Bayesian decision rule requires only a modified threshold $\theta = \frac{\pi}{1-\pi}\frac{1-\tilde{\pi}}{\tilde{\pi}}$ rather than the standard threshold of 1. The practical innovation develops three empirical classifiers (ALS, CLS, CLS-EM) incorporating label shift correction into the VAE-PU-Bayes framework, along with Direct and EM estimators for the shifted prior based on the relationship $\tilde{\pi} = \frac{P(\tilde{S}=1)}{P(S=1)} \times \pi$. Experiments across synthetic and real-world datasets with varying shift scenarios show the CLS estimator consistently outperforms alternatives, demonstrating robustness across different labeling biases. This establishes a mathematical framework for understanding how distribution shifts affect partially labeled data and provides practical tools for detecting and correcting label shift for augmented PU data.

Thesis importance and impact. This research makes significant contributions to both the theoretical foundations and practical applications of PU learning. The work addresses critical gaps in the field by establishing theoretical results about scenario compatibility, developing novel generative approaches for biased labeling, introducing new frameworks for quality control and augmented prediction scenarios, and providing solutions for distribution shift challenges. The proposed methods consistently demonstrate substantial empirical improvements across diverse datasets and evaluation metrics, with significant performance gains over existing competing approaches. Most notably, VAE-PU-Bayes emerges as the strongest and most versatile method developed in this thesis, consistently delivering superior performance across all experimental settings and establishing new performance benchmarks for single sample no-SCAR PU learning. Beyond algorithmic contributions, the thesis establishes new research paradigms that extend the applicability of PU learning to previously unexplored scenarios, provides mathematical frameworks for understanding complex relationships in partially labeled data, and offers practical tools that are applicable to real-world problems in domains such as medical diagnosis, fraud detection, and recommendation systems. The comprehensive experimental validation across synthetic and real-world datasets, combined with rigorous theoretical analysis, positions this work to significantly influence future research directions in PU learning and related areas of machine learning with incomplete supervision.

8.2 Future research avenues

Several promising directions emerge from this research, spanning both foundational theoretical work and practical algorithmic improvements. To begin with, the scenario compatibility analysis could be extended to more types of methods and different data characteristics, with particular focus on identifying which case control methods can be easily adapted to single sample settings. Understanding the fundamental differences between these scenarios more deeply could help researchers avoid common pitfalls and develop more robust methods that work across different data collection paradigms.

A fundamental challenge across all VAE-PU-based methods lies in accurately identifying the true positive examples within the unlabeled dataset. While VAE-PU+OCC and VAE-PU-Bayes represent significant advances, the quality of the generated positive unlabeled set remains a critical bottleneck. Future research could focus on developing more sophisticated generative models beyond variational autoencoders, such as GANs or diffusion models. Additionally, developing hybrid approaches that combine multiple detection strategies could yield more robust identification of positive unlabeled examples.

In PU learning class imbalance (controlled by π) and labeling intensity (controlled by c and propensity function $e(x)$) are often confounded. Understanding how these factors interact represents an important challenge. Future research might focus on developing frameworks for systematically studying these effects in isolation. This could reveal whether certain algorithms are more sensitive to class imbalance versus labeling bias, leading to more targeted algorithm selection strategies.

Throughout this thesis, the class prior π has been assumed known, which is a limitation in practical applications. Developing robust estimators for π specifically within the VAE-PU-Bayes framework presents unique opportunities. Future research could explore joint estimation of the class prior alongside posterior probabilities and investigate sensitivity to misspecification of π.

The FOR control framework would benefit from procedures robust to similar inlier and outlier distributions, as well as incorporation of class prior estimators such as Storey's estimator. The current method's sensitivity to outlier score quality suggests that developing more robust procedures could significantly expand its applicability.

The augmented PU learning paradigm, as a new field, presents rich opportunities for further development. Improving the modeling of $y(x)$ to be more sensitive to corrections via the d_B^{PU} rule represents an important challenge. Developing new classifiers that estimate both $y(x)$ and $s(x)$ directly, similar to LBE but optimized for the augmented setting, could yield significant advances. The exceptional performance of VAE-PU-Bayes also motivates incorporating the d_B^{PU} rule into additional PU learning frameworks beyond the VAE-based approach.

While this thesis addresses label shift in the augmented PU setting, it did not touch upon the more common scenario of label shift in standard PU learning. Developing methods to detect and estimate label shift when only (X, S) information is available presents unique challenges, as the information present in such scenario is more limited – this is evident by low number of papers considering label shift in no-SCAR case. Additionally, developing more stable label shift prior estimators and allowing for different propensity scores between training and target datasets would handle more complex real-world scenarios.

Table of symbols

General Notation

X	Feature vector, $X \in \mathbb{R}^p$
Y	True class indicator, $Y \in \{-1, 1\}$
S	Labeling indicator, $S \in \{0, 1\}$
π	Class prior probability, $\pi = P(Y = 1)$
c	Label frequency, $c = P(S = 1\|Y = 1)$
$e(x)$	Propensity score, $e(x) = P(S = 1\|Y = 1, X = x)$
$y(x)$	Posterior probability of positive class, $y(x) = P(Y = 1\|X = x)$
$s(x)$	Posterior probability of being labeled, $s(x) = P(S = 1\|X = x)$
$g(x)$	Classification function
$d(x)$	Classifier, $d(x) = 2\mathbb{I}\{g(x) \geq 0\} - 1$
$l(\cdot, \cdot)$	Loss function
$R(g)$	Risk function
$\mathscr{D}$	Training dataset
n	Number of training examples
χ_{PL}	Positive labeled set
χ_{PU}	Positive unlabeled set
χ_U	Unlabeled set
χ_N	Negative set
L	Labeled set
U	Unlabeled set
$\mathbb{I}\{\cdot\}$	Indicator function
$\mathbb{E}$	Expectation operator
$\mathbb{R}$	Set of real numbers
$\perp\!\!\!\perp$	Statistically independence
$\not\perp\!\!\!\perp$	Not independent
$\mathrm{d}\cdot$	Differential operator

diag	Diagonal matrix operator

Chapter 2: Preliminaries

P_{XY}	Joint probability distribution of (X, Y)
$P_{X\|Y=1}$	Probability distribution of positive examples
$P_{X\|Y=-1}$	Probability distribution of negative examples
P_{XYS}	Joint probability distribution of (X, Y, S)
$l_{0-1}(t)$	0-1 loss function
$l_{\text{log}}(t)$	Logistic loss function
$l_{\text{inv}}(t)$	Inverse sigmoid loss function
$l_{\text{exp}}(t)$	Exponential loss function
$l_{\text{sig}}(t)$	Sigmoid loss function
$g^*(x)$	Optimal classification function
$g^*_{\text{log}}(x)$	Optimal classification function for logistic loss
$g^*_{\text{exp}}(x)$	Optimal classification function for exponential loss
$P_{X\|S=1}$	Distribution of features given labeled status
$P_{X\|S=0}$	Distribution of features given unlabeled status

Chapter 3: PU Learning Scenarios

n_L	Number of labeled examples
n_U	Number of unlabeled examples
$\widehat{R}_{\text{uPU}}$	Empirical unbiased PU risk
$\widehat{R}_{\text{nnPU}}$	Empirical non-negative PU risk
$\widehat{R}_{\text{uPU}_{\text{CC}}}$	Empirical unbiased PU risk (case control)
$\widehat{R}_{\text{nnPU}_{\text{CC}}}$	Empirical non-negative PU risk (case control)
$\widehat{R}_{\text{uPU}_{\text{SS}}}$	Empirical unbiased PU risk (single sample)
$\widehat{R}_{\text{nnPU}_{\text{SS}}}$	Empirical non-negative PU risk (single sample)
R^L	Labeled risk component
R^D	General distribution risk component
R^{corr}	PU SCAR correction term
β	Hyperparameter for risk truncation
γ	Step size discount factor
P_U	Distribution of unlabeled data

SS	Single sample scenario
CC	Case control scenario

Chapter 4: VAE-PU Methods

z	Latent space representation
$\widehat{x}$	Reconstructed input
μ	Mean parameter of latent distribution
σ	Standard deviation parameter of latent distribution
ϵ	Random noise from standard normal distribution
$q(z\|x)$	Variational approximation to posterior
$p(z)$	Prior distribution
D_{KL}	Kullback-Leibler divergence
$\mathscr{L}_{\text{ELBO}}$	Evidence Lower Bound
$\widehat{\mathscr{L}_{\text{ELBO}}}$	Empirical Evidence Lower Bound
K	Number of clusters
c	Cluster indicator
π_k	Mixing coefficient for cluster k
h_y	Latent class assignment representation
h_s	Latent labeling status representation
f_θ	Decoder network with parameters θ
h_ϕ	Encoder network with parameters ϕ
D	Discriminator network
$\mathscr{L}_{\text{adv}}$	Adversarial generation loss
$\mathscr{L}_{\text{target}}$	Target loss
α, β	Hyperparameters for loss weighting
$\widetilde{\chi}_{PU}$	Generated positive unlabeled set
$\widetilde{\mu}$	Encoder mean output
$\widetilde{\sigma}$	Encoder variance output
μ_x, σ_x	Decoder output parameters
r_s	Observation classifier output
f_{s_ξ}	Observation classifier network
γ_c	Shorthand for $q(c\|x)$
J	Dimensionality of latent variables
J_y	Dimensionality of class latent variable
J_s	Dimensionality of labeling latent variable

L	Number of Monte Carlo samples
p	Dimensionality of input data

Chapter 5: False Omission Rate Control

$\widehat{s}$	Score statistic
F	Cumulative distribution function
p_i	p-value for observation i
F_1	CDF of p-values for outliers
$G(t)$	Mixture distribution of p-values
R	Number of rejected null hypotheses
V	Number of falsely rejected nulls
Z	Number of falsely not rejected alternatives
NR	Number of not rejected items
FOR	False Omission Rate
BFOR	Bayesian False Omission Rate
FDR	False Discovery Rate
u	Threshold for p-values
u^*	Optimal threshold
$p_{(i)}$	i-th order statistic of p-values
$\mathscr{D}^{\text{train}}$	Training set
$\mathscr{D}^{\text{cal}}$	Calibration set
p_{single}	Single-split p-value
p_{med}	Median p-value over multiple splits
p_{2med}	Doubled median p-value
$H_{0,i}$	Null hypothesis for observation i
$H_{1,i}$	Alternative hypothesis for observation i
k	Number of random splits in Multisplit
α	Control level
δ	Probability parameter

Chapter 6: Augmented PU Learning

$\widetilde{y}(x,s)$	Posterior probability given augmented predictors
$d_B^{PU}(x,s)$	Bayes rule for augmented PU data

$d_B(x)$	Standard Bayes rule
$\eta(x)$	Posterior probability on unlabeled stratum
$w(x)$	Auxiliary function, $w(x) = 1 + s(x) - 2y(x)$
L^*_{PU}	Bayes risk for augmented PU
L^{*0}_{PU}	Bayes risk on unlabeled stratum
$\Delta(d)$	Excess risk of classifier d
$\mathrm{OR}(x)$	Odds ratio
$\mathrm{OD}(x)$	Odds of positive class
$\Phi(\cdot)$	Standard normal CDF
$\phi(\cdot)$	Standard normal PDF
$\sigma(\cdot)$	Sigmoid function
U-ACC	Accuracy on unlabeled stratum
U-metrics	Metrics calculated on unlabeled stratum
L^*	Bayes risk
μ	Mean parameter in examples

Chapter 7: Label Shift

$\widetilde{X}$	Feature vector in shifted distribution
$\widetilde{Y}$	Class indicator in shifted distribution
$\widetilde{S}$	Labeling indicator in shifted distribution
$\widetilde{\pi}$	Class prior in shifted distribution
$\widetilde{c}$	Label frequency in shifted distribution
$\widetilde{y}(x)$	Posterior probability in shifted distribution
$\widetilde{s}(x)$	Labeling probability in shifted distribution
$\widetilde{e}(x)$	Propensity score in shifted distribution
$\widetilde{\eta}(x)$	Posterior probability on unlabeled stratum (shifted)
$\widetilde{d}^{PU}_B$	Bayes rule for label-shifted augmented PU
θ	Threshold parameter for label shift
$\widehat{\widetilde{\pi}}$	Direct estimator of shifted prior
$\widehat{\widetilde{\pi}}_{EM}$	EM estimator of shifted prior
γ	Prior probability of being labeled, $\gamma = P(S = 1)$
$\widetilde{\gamma}$	Prior probability of being labeled (shifted), $\widetilde{\gamma} = P(\widetilde{S} = 1)$
N_1	Number of labeled examples in training set
M_1	Number of labeled examples in test set
m	Number of test examples

IR	Imbalance ratio between datasets
$I(\mathscr{D})$	Imbalance measure of dataset $\mathscr{D}$
$P_{\widetilde{X}\widetilde{Y}}$	Joint distribution for shifted data
$P_{\widetilde{X}\widetilde{Y}\widetilde{S}}$	Joint distribution for shifted augmented data
$\widetilde{\mathscr{D}}$	Shifted dataset
f_X	Density or probability mass function of X
ψ	Auxiliary parameter in odds ratio calculation

Evaluation Metrics

TP	True positives
TN	True negatives
FP	False positives
FN	False negatives
Accuracy	Classification accuracy
Precision	Precision metric
Recall	Recall metric
F1 score	F1 score metric
U-ACC	Accuracy on unlabeled stratum
SEM	Standard error of the mean
MSE	Mean squared error
Balanced Accuracy	Balanced accuracy metric
U-Balanced Accuracy	Balanced accuracy on unlabeled stratum

Methods and Algorithms

SCAR	Selected Completely At Random assumption
SAR	Selected At Random assumption
uPU	Unbiased PU learning
nnPU	Non-negative PU learning
VAE-PU	Variational autoencoder for PU learning
VAE-PU+OCC	VAE-PU with One-Class Classification
VAE-PU+FOR	VAE-PU with False Omission Rate control
VAE-PU-Bayes	VAE-PU with Bayesian decision rule
SAR-EM	Selected At Random Expectation Maximization

LBE	Learning with Biased Examples for Feature-Dependent Labeling
OCC	One-Class Classification
VaDE	Variational Deep Embedding
A^3	Activation Anomaly Analysis
ECOD	Empirical cumulative distribution outlier detection
OC-SVM	One-Class Support Vector Machine
IForest	Isolation Forest
CLS	Cut-off label shift estimator
CLS-EM	CLS with EM estimation
ALS	Augmented label shift estimator

Bibliography

Bank, D., Koenigstein, N., and Giryes, R. (Jan. 1, 2023). „Autoencoders". In: *Machine Learning for Data Science Handbook*. Springer International Publishing, pp. 353–374. ISBN: 978-3-031-24627-2.

Bates, S., Candès, E., Lei, L., Romano, Y., and Sesia, M. (2023). „Testing for outliers with conformal p-values". In: *The Annals of Statistics* 51.1. Publisher: Institute of Mathematical Statistics, pp. 149–178.

Bekker, J. and Davis, J. (Apr. 2018a). „Estimating the class prior in positive and unlabeled data through decision tree induction". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 32.1, pp. 2712–2719. URL: `https://ojs.aaai.org/index.php/AAAI/article/view/11715`.

— (Sept. 10, 2018b). „Learning from Positive and Unlabeled Data under the Selected At Random Assumption". In: *Proceedings of the Second International Workshop on Learning with Imbalanced Domains: Theory and Applications*. Vol. 94. Proceedings of Machine Learning Research. PMLR, pp. 8–22. URL: `https://proceedings.mlr.press/v94/bekker18a.html`.

— (Apr. 2020). „Learning from positive and unlabeled data: a survey". In: *Machine Learning* 109.4. Publisher: Springer Science and Business Media LLC, pp. 719–760.

Bekker, J., Robberechts, P., and Davis, J. (2019). „Beyond the selected completely at random assumption for learning from positive and unlabeled data". In: *Proceedings of the 2019 European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*. Springer, Cham, pp. 71–85. ISBN: 978-3-030-46147-8.

Benjamini, Y. and Hochberg, Y. (1995). „Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 57, pp. 289–300.

Berthon, A., Han, B., Niu, G., Liu, T., and Sugiyama, M. (2021). „Confidence scores make instance-dependent label-noise learning possible". In: *International Conference on Machine Learning*. PMLR, pp. 825–836.

Bishop, C. (2006). *Pattern Recognition and Machine Learning*. New York, NY: Springer.

Blanchard, G., Lee, G., and Scott, C. (2010). „Semi-Supervised Novelty Detection". In: *Journal of Machine Learning Research* 11.99, pp. 2973–3009. URL: `http://jmlr.org/papers/v11/blanchard10a.html`.

Bühlmann, P. and Geer, S. van de (2011). *Statistics for High-Dimensional Data*. Springer.

Calvo, B., Larranaga, P., and Lozano, J. (Dec. 2007). „Learning Bayesian classifiers from positive and unlabeled examples". In: *Pattern Recognition Letters* 28, pp. 2375–2384.

Chaudhari, S. and Shevade, S. (2012). „Learning from positive and unlabelled examples using maximum margin clustering". In: *Proceedings of the 19th International Conference on Neural Information Processing - Volume Part III*. ICONIP'12. event-place: Doha, Qatar. Berlin, Heidelberg: Springer-Verlag, pp. 465–473. ISBN: 978-3-642-34486-2. URL: `https://doi.org/10.1007/978-3-642-34487-9_56`.

Chen, J.-L., Cai, J.-J., Jiang, Y., and Huang, S.-J. (Oct. 2021). „PU Active Learning for Recommender Systems". In: *Neural Processing Letters* 53, pp. 1–14.

Claesen, M., De Smet, F., Suykens, J. A., and De Moor, B. (July 2015). „A robust ensemble approach to learn from positive and unlabeled data using SVM base models". In: *Neurocomputing* 160. Publisher: Elsevier BV, pp. 73–84.

Coudray, O., Keribin, C., Massart, P., and Pamphile, P. (2023). „Risk Bounds for positive-unlabeled learning under the selected at random assumption". In: *Journal of Machine Learning Research* 24, pp. 1–31.

Cover, T. and Thomas, J. (1991). *Elements of Information Theory*. New York, NY: Wiley.

DeGroot, M. (2004). *Optimal Statistical Decisions*. Wiley Classics Library. Wiley. ISBN: 978-0-471-68029-1. URL: `https://books.google.pl/books?id=7rDY2_r4bmEC`.

Denis, F., Gilleron, R., and Letouzey, F. (2005). „Learning from positive and unlabeled examples". In: *Theoretical Computer Science* 348.1, pp. 70–83. ISSN: 0304-3975. URL: `https://www.sciencedirect.com/science/article/pii/S0304397505005256`.

Denis, F., Laurent, A., and Tommasi, M. (Jan. 2003). „Text classification and co-training from positive and unlabeled examples". In: *Proceedings of the ICML 2003 Workshop: The Continuum from Labeled to Unlabeled Data*.

Dudoit, S. and Laan, M. J. v. d. (Jan. 2008). *Multiple Testing Procedures with Applications to Genomics*. Publication Title: International Statistical Review. Springer. 309–310.

Efron, B. (2010). *Large-Scale Inference: Empirical Bayes Methods for Estimation, Testing, and Prediction*. Institute of Mathematical Statistics Monographs. Cambridge University Press.

Elkan, C. (2001). „The foundations of cost-sensitive learning". In: *In Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence*, pp. 973–978.

Elkan, C. and Noto, K. (Aug. 2008). „Learning classifiers from only positive and unlabeled data". In: *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 213–220.

Furmańczyk, K., Mielniczuk, J., Rejchel, W., and Teisseyre, P. (Sept. 2022). „Joint estimation of posterior probability and propensity score function for positive and unlabelled data". In: *arXiv preprint arXiv:2209.07787*.

— (2023). „Double Logistic Regression Approach to Biased Positive-Unlabeled Data". In: *Proceedings of the European Conference on Artificial Intelligence*. ECAI'23, pp. 764–771.

Garg, S., Wu, Y., Balakrishnan, S., and Lipton, Z. C. (2020). „A unified view of label shift estimation". In: *Proceedings of the 34th International Conference on Neural Information Processing Systems*. NIPS' 20, pp. 1–11.

Genovese, C. and Wasserman, L. (2002). „Operating characteristics and extensions of False Discovery Rate procedures". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 64.3, pp. 499–517.

Gong, C., Wang, Q., Liu, T., Han, B., You, J. J., Yang, J., and Tao, D. (2021). „Instance-Dependent Positive and Unlabeled Learning with Labeling Bias Estimation". In: *IEEE transactions on pattern analysis and machine intelligence* PP.

González, P., Castaño, A., Chawla, N., and Coz, J. (2017). „A Review on Quantification Learning". In: *ACM Comput. Surv.* 50.5.

Guyon, I. (2004). *Madelon*. Published: UCI Machine Learning Repository.

Harvey, C. and Liu, Y. (Jan. 2017). „False (and Missed) Discoveries in Financial Economics". In: *SSRN Electronic Journal*.

Hastie, T., Tibshirani, R., and Wainwright, M. (May 2015). *Statistical Learning with Sparsity: The Lasso and Generalizations*. Chapman and Hall/CRC. 1–337. ISBN: 978-0-429-17158-1.

Hayashi, T., Cimr, D., Fujita, H., and Cimler, R. (2024). *Critical Review for One-class Classification: recent advances and the reality behind them*. _eprint: 2404.17931. URL: `https://arxiv.org/abs/2404.17931`.

He, F., Liu, T., Webb, G. I., and Tao, D. (2018). „Instance-Dependent PU Learning by Bayesian Optimal Relabeling". In: *CoRR* abs/1808.02180. arXiv: `1808.02180`. URL: `http://arxiv.org/abs/1808.02180`.

Hsieh, C.-J., Natarajan, N., and Dhillon, I. S. (2015). „PU learning for matrix completion". In: *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*. ICML'15. Place: Lille, France. JMLR.org, pp. 2445–2453.

Iyer, A., Nath, S., and Sarawagi, S. (2014). „Maximum mean discrepancy for class ratio estimation: convergence bounds and kernel selection". In: *Proceedings of the 31th International Conferencce on Machine Learning*. IMLR W & CP vol. 32.

Ji, X., Henriques, J. F., and Vedaldi, A. (2019). „Invariant information clustering for unsupervised image classification and segmentation". In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 9865–9874.

Jiang, Z., Zheng, Y., Tan, H., Tang, B., and Zhou, H. (2017). „Variational Deep Embedding: An Unsupervised and Generative Approach to Clustering". In: *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pp. 1965–1972. URL: `https://doi.org/10.24963/ijcai.2017/273`.

Kato, M., Teshima, T., and Honda, J. (2019). „Learning from Positive and Unlabeled Data with a Selection Bias". In: *International Conference on Learning Representations*.

Ke, T., Yang, B., Zhen, L., Tan, J., Li, Y., and Jing, L. (2012). „Building high-performance classifiers using positive and unlabeled examples for text classification". In: *International Symposium on Neural Networks*. Springer, pp. 187–195.

Kingma, D. P. and Welling, M. (2019). „An Introduction to Variational Autoencoders". In: *Foundations and Trends® in Machine Learning* 12.4. Publisher: Now Publishers, pp. 307–392. ISSN: 1935-8245. URL: `http://dx.doi.org/10.1561/2200000056`.

Kiryo, R., Niu, G., Plessis, M. C. du, and Sugiyama, M. (2017). „Positive-Unlabeled Learning with Non-Negative Risk Estimator". In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. NIPS'17. event-place: Long Beach, California, USA. Red Hook, NY, USA: Curran Associates Inc., pp. 1674–1684. ISBN: 978-1-5108-6096-4.

Kögler, K., Shevchenko, A., Hassani, H., and Mondelli, M. (2024). „Compression of structured data with autoencoders: provable benefit of nonlinearities and depth". In: *Proceedings of the 41st International Conference on Machine Learning*. ICML'24. Place: Vienna, Austria. JMLR.org.

Lee, W. and Liu, B. (2003). „Learning with positive and unlabeled examples using weighted logistic regression". In: *Proceedings of the Twentieth International Conference on Machine Learning*, pp. 448–455.

Lee, W.-H., Ozger, M., Challita, U., and Sung, K. W. (Sept. 2021). „Noise Learning-Based Denoising Autoencoder". In: *IEEE Communications Letters* 25.9. Publisher: Institute of Electrical and Electronics Engineers (IEEE), pp. 2983–2987. ISSN: 2373-7891. URL: `http://dx.doi.org/10.1109/LCOMM.2021.3091800`.

Lehmann, E. and Romano, J. P. (2022). *Testing Statistical Hypotheses*. Springer Texts in Statistics. Cham: Springer International Publishing. ISBN: 978-3-030-70577-0 978-3-030-70578-7. URL: `https://link.springer.com/10.1007/978-3-030-70578-7` (visited on 09/02/2025).

Li, X. and Liu, B. (2003). „Learning to Classify Texts Using Positive and Unlabeled Data". In: *Proceedings of the 18th International Joint Conference on Artificial Intelligence*. IJCAI'03, pp. 587–592.

— (2005). „Learning from Positive and Unlabeled Examples with Different Data Distributions". In: *Machine Learning: ECML 2005*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 218–229. ISBN: 978-3-540-31692-3.

Li, Z., Zhao, Y., Hu, X., Botta, N., Ionescu, C., and Chen, G. H. (2022). „ECOD: Unsupervised Outlier Detection Using Empirical Cumulative Distribution Functions". In: *IEEE Transactions on Knowledge and Data Engineering*.

Liang, Q., Zhu, M., Wang, Y., Wang, X., Zhao, W., Yang, M., Wei, H., Han, B., and Zheng, X. (June 2023). „Positive Distribution Pollution: Rethinking Positive Unlabeled Learning from a Unified Perspective". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 37.7, pp. 8737–8745. URL: `https://ojs.aaai.org/index.php/AAAI/article/view/26051`.

Lipton, Z. C., Wang, Y., and Smola, A. J. (2018). „Detecting and Correcting for Label Shift with Black Box Predictors". In: *Proceedings of the 35th International Conference on Machine Learning*. ICML' 18, pp. 3128–3136.

Liu, B., Dai, Y., Li, X., Lee, W. S., and Yu, P. S. (2003). „Building Text Classifiers Using Positive and Unlabeled Examples". In: *Proceedings of the Third IEEE International Conference on Data Mining*. ICDM '03, pp. 179–186.

Liu, B., Lee, W. S., Yu, P. S., and Li, X. (2002). „Partially Supervised Classification of Text Documents". In: *Proceedings of the 19-th International Conference on Machine Learning*. ICLM'02, pp. 387–394.

Liu, F. T., Ting, K. M., and Zhou, Z.-H. (2008). „Isolation Forest". In: *2008 Eighth IEEE International Conference on Data Mining*, pp. 413–422.

Liu, R. Y., Parelius, J. M., and Singh, K. (1999). „Multivariate analysis by data depth: descriptive statistics, graphics and inference". In: *The Annals of Statistics* 27.3. Publisher: Institute of Mathematical Statistics, pp. 783–858. URL: `https://doi.org/10.1214/aos/1018031260`.

Liu, Z., Shi, W., Li, D., and Qin, Q. (2005). „Partially supervised classification – based on weighted unlabeled samples support vector machine". In: *Advanced Data Mining and Applications*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 118–129.

Maslej, N., Fattorini, L., Perrault, R., Parli, V., Reuel, A., Brynjolfsson, E., Etchemendy, J., Ligett, K., Lyons, T., Manyika, J., Niebles, J. C., Shoham, Y., Wald, R., and Clark, J. (Apr. 2024). *The AI Index 2024 Annual Report*. Stanford, CA: AI Index Steering Committee, Institute for Human-Centered AI, Stanford University.

Meinshausen, N., Meier, L., and Bühlmann, P. (Nov. 2008). „P-values for high-dimensional regression". In: *Journal of the American Statistical Association* 104, pp. 1671–1681.

Menon, A., Rooyen, B., Ong, C. S., and Williamson, B. (2015). „Learning from Corrupted Binary Labels via Class-Probability Estimation". In: *Proceedings of the 32nd International Conference on Machine Learning*. Proceedings of Machine Learning Research, pp. 125–134.

Mielniczuk, J. and Wawrzeńczyk, A. (Sept. 2023). „One-Class Classification Approach to Variational Learning from Biased Positive Unlabeled Data". In: *ECAI 2023*. ISSN: 1879-8314. IOS Press. ISBN: 978-1-64368-437-6. URL: `http://dx.doi.org/10.3233/FAIA230457`.

— (Oct. 2024). „Augmented Prediction of a True Class for Positive Unlabeled Data Under Selection Bias". In: *ECAI 2024*. ISSN: 1879-8314. IOS Press. ISBN: 978-1-64368-548-9. URL: `http://dx.doi.org/10.3233/FAIA240806`.

— (2025a). „Accounting for label shift of Positive Unlabeled data under selection bias". In: *International Journal of Applied Mathematics and Computer Science* 35.3, pp. 507–517.

— (May 2025b). „Single-sample Versus Case-control Sampling Scheme for Positive Unlabeled Data: the Story of Two Scenarios". In: *Fundamenta Informaticae* 193, pp. 29–45.

Mordelet, F. and Vert, J.-P. (Oct. 2010). „A bagging SVM to learn from positive and unlabeled examples". In: *Pattern Recognition Letters* 37.1, pp. 201–209.

Moya, M. M., Koch, M. W., and Hostetler, L. D. (Jan. 1993). „One-class classifier networks for target recognition applications". In: *NASA STI/Recon Technical Report* N 93, p. 24043.

Na, B., Kim, H., Song, K., Joo, W., Kim, Y.-Y., and Moon, I.-C. (2020). „Deep Generative Positive-Unlabeled Learning under Selection Bias". In: *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. CIKM '20. event-place: Virtual Event, Ireland. New York, NY, USA: Association for Computing Machinery, pp. 1155–1164. ISBN: 978-1-4503-6859-9.

Nguyen, H. H., Nguyen, C. N., Dao, X. T., Duong, Q. T., Kim, D. P. T., and Pham, M.-T. (2024). „Variational Autoencoder for Anomaly Detection: A Comparative Study". In: *ArXiv* abs/2408.13561. URL: `https://api.semanticscholar.org/CorpusID:271957110`.

Northcutt, C. G., Wu, T., and Chuang, I. L. (2017). *Learning with Confident Examples: Rank Pruning for Robust Classification with Noisy Labels*. _eprint: 1705.01936. URL: `https://arxiv.org/abs/1705.01936`.

Ortega Vázquez, C., Broucke, S. vanden, and De Weerdt, J. (Mar. 2023). „Hellinger distance decision trees for PU learning in imbalanced data sets". In: *Mach. Learn.* 113.7. Place: USA Publisher: Kluwer Academic Publishers, pp. 4547–4578. ISSN: 0885-6125. URL: `https://doi.org/10.1007/s10994-023-06323-y`.

Park, J. W., Chu, M. K., Kim, J. M., Park, S. G., and Cho, S. J. (2016). „Analysis of Trigger Factors in Episodic Migraineurs Using a Smartphone Headache Diary Applications". In: *PloS one* 11.2, pp. 1–13.

Pimentel, M. A. F., Clifton, D. A., Clifton, L., and Tarassenko, L. (2014). „A review of novelty detection". In: *Signal Processing* 99, pp. 215–249. ISSN: 0165-1684.

Plessis, M. du, Niu, G., and Sugiyama, M. (July 7, 2015). „Convex Formulation for Learning from Positive and Unlabeled Data". In: *Proceedings of the 32nd International Conference on*

Machine Learning. Vol. 37. Proceedings of Machine Learning Research. Lille, France: PMLR, pp. 1386–1394. URL: `https://proceedings.mlr.press/v37/plessis15.html`.

Plessis, M. C. du, Niu, G., and Sugiyama, M. (2014). „Analysis of Learning from Positive and Unlabeled Data". In: *Advances in Neural Information Processing Systems*. Vol. 27. Curran Associates, Inc., pp. 703–711.

Płatek, M. and Mielniczuk, J. (2023). „Enhancing naive classifier for positive unlabeled data based on logistic regression approach". In: *Proceedings of the 18th Conference on Computer Science and Intelligence Systems*. ACSIS. Vol. 35, pp. 225–233.

Rejchel, W., Teisseyre, P., and Mielniczuk, J. (2024). „Joint empirical risk minimization for instance-dependent positive-unlabeled data". In: *Knowledge-Based Systems* 304, p. 112444. ISSN: 0950-7051. URL: `https://www.sciencedirect.com/science/article/pii/S0950705124010785`.

Ruff, L., Kauffmann, J., Vandermeulen, R., Montavon, G., Samek, W., Kloft, M., Dietterich, T., and Müller, K. (Feb. 2021). „A Unifying Review of Deep and Shallow Anomaly Detection". In: *Proceedings of the IEEE* PP, pp. 1–40.

Sadok, S., Leglaive, S., Girin, L., Alameda-Pineda, X., and Séguier, R. (2024). „A multimodal dynamical variational autoencoder for audiovisual speech representation learning". In: *Neural Networks* 172, p. 106120. ISSN: 0893-6080. URL: `https://www.sciencedirect.com/science/article/pii/S0893608024000340`.

Saerens, M., Latinne, P., and Decaestecker, C. (2002). „Adjusting the outputs of a classifier to new a priori probabilities: a simple procedure". In: *Neural Comput.* 14.1, pp. 21–41.

Schlegl, T., Seeböck, P., Waldstein, S. M., Schmidt-Erfurth, U., and Langs, G. (2017). „Unsupervised Anomaly Detection with Generative Adversarial Networks to Guide Marker Discovery". In: *Information Processing in Medical Imaging*. Springer, pp. 146–157.

Schölkopf, B., Platt, J., Shawe-Taylor, J., Smola, A., and Williamson, R. (July 2001). „Estimating Support of a High-Dimensional Distribution". In: *Neural Computation* 13, pp. 1443–1471.

Sellamanickam, S., Garg, P., and Selvaraj, S. K. (2011). „A pairwise ranking based approach to learning with positive and unlabeled examples". In: *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*. CIKM '11. event-place: Glasgow, Scotland, UK. New York, NY, USA: Association for Computing Machinery, pp. 663–672. ISBN: 978-1-4503-0717-8. URL: `https://doi.org/10.1145/2063576.2063675`.

Shaker, A., Maaz, M., Abdul Rasheed, H., Khan, S., Yang, M.-H., and Khan, F. (Oct. 2023). „SwiftFormer: Efficient Additive Attention for Transformer-based Real-time Mobile Vision Applications". In: pp. 17379–17390.

Shao, Y.-H., Chen, W.-J., Liu, L.-M., and Deng, N.-Y. (2015). „Laplacian unit-hyperplane learning from positive and unlabeled examples". In: *Information Sciences* 314, pp. 152–168.

ISSN: 0020-0255. URL: `https://www.sciencedirect.com/science/article/pii/S002002551500239X`.

Sperl, P., Schulze, J.-P., and Böttinger, K. (2021). „Activation Anomaly Analysis". In: *Machine Learning and Knowledge Discovery in Databases*. Springer International Publishing, pp. 69–84.

Steinberg, D. and Cardell, N. (Jan. 1992). „Estimating logistic regression models when the dependent variable has no variance". In: *Communications in Statistics-theory and Methods - COMMUN STATIST-THEOR METHOD* 21, pp. 423–450.

Storey, J. (2002). „Direct approach to false discovery rates". In: *Journal of the Royal Statistical Society. Series B (Methodological)* 64. Publisher: [Royal Statistical Society, Wiley], pp. 479–498.

Su, G., Chen, W., and Xu, M. (Aug. 2021). „Positive-Unlabeled Learning from Imbalanced Data". In: pp. 2995–3001.

Sugiyama, M., Bao, H., Ishida, T., Lu, N., Sakai, T., and ang, N. (2022). *Machine Learning from Weak Supervision*. MIT Press.

Takahashi, H., Ichinose, N., and Yasusei, O. (2022). „False-negative rate of SARS-CoV-2 RT-PCR tests and its relationship to test timing and illness severity". In: *IdCases* 28.

Tang, X., Xu, C., Tao, H., Ma, X., and Hou, C. (2025). „Confidence-Based PU Learning With Instance-Dependent Label Noise". In: *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–15.

Tang, Z., Pei, S., Zhang, Z., Zhu, Y., Zhuang, F., Hoehndorf, R., and Zhang, X. (July 2022). „Positive-Unlabeled Learning with Adversarial Data Augmentation for Knowledge Graph Completion". In: *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*. IJCAI-2022. International Joint Conferences on Artificial Intelligence Organization, pp. 2248–2254. URL: `http://dx.doi.org/10.24963/ijcai.2022/312`.

Vaz, A., Izbicki, R., and Stern, R. (2019). „Quantification Under Prior Probability Shift: the Ratio Estimator and its Extensions". In: *Journal of Machine Learning Research* 20, pp. 1–33.

Vovk, V., Gammerman, A., and Shafer, G. (Jan. 2005). „Algorithmic Learning in a Random World". In: *Algorithmic Learning in a Random World*. New York, NY: Springer Science & Business Media.

Vovk, V., Gammerman, A., and Saunders, C. (June 1999). „Machine-Learning Applications of Algorithmic Randomness". In: *Proceedings of the Sixteenth International Conference on Machine Learning*. ICML '99. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., pp. 444–453. ISBN: 1-55860-612-2.

Wang, W., Wei, F., Dong, L., Bao, H., Yang, N., and Zhou, M. (2020). „MINILM: Deep Self-Attention Distillation for Task-Agnostic Compression of Pre-Trained Transformers". In: *Proceedings of the 34th International Conference on Neural Information Processing Systems*.

NIPS'20. event-place: Vancouver, BC, Canada. Red Hook, NY, USA: Curran Associates Inc., pp. 5776–5788.

Wang, W., Gan, Z., Xu, H., Zhang, R., Wang, G., Shen, D., Chen, C., and Carin, L. (June 2019). „Topic-Guided Variational Auto-Encoder for Text Generation". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 166–177. URL: https://aclanthology.org/N19-1015/.

Ward, G., Hastie, T., Barry, S., Elith, J., and Leathwick, J. (2009). „Presence-only data and the EM algorithm". In: *Biometrics* 65, pp. 554–563.

Wawrzeńczyk, A. and Mielniczuk, J. (2022). „Strategies for fitting logistic regression for positive and unlabeled data revisited". In: *International Journal of Applied Mathematics and Computer Science* 4.2, pp. 299–309.

— (2023a). „One-class classification approach to variational learning from biased positive unlabelled data". In: *Proceedings of the European Conference on Artificial Intelligence*. ECAI'23, pp. 1720–1727.

— (2023b). „Outlier Detection Under False Omission Rate Control". In: *Computational Science – ICCS 2023*. ISSN: 1611-3349. Springer Nature Switzerland, pp. 610–625. ISBN: 978-3-031-36024-4. URL: http://dx.doi.org/10.1007/978-3-031-36024-4_47.

Xu, Z., Qi, Z., and Zhang, J. (Nov. 1, 2014). „Learning with positive and unlabeled examples using biased twin support vector machine". In: *Neural Computing and Applications* 25.6, pp. 1303–1311. ISSN: 1433-3058. URL: https://doi.org/10.1007/s00521-014-1611-3.

Yang, X., Song, Z., King, I., and Xu, Z. (Sept. 2023). „A Survey on Deep Semi-Supervised Learning". In: *IEEE Transactions on Knowledge and Data Engineering* 35.9. Publisher: Institute of Electrical and Electronics Engineers (IEEE), pp. 8934–8954. ISSN: 2326-3865. URL: http://dx.doi.org/10.1109/TKDE.2022.3220219.

Ye, C., Tsuchida, R., Petersson, L., and Barnes, N. (2024). „Label Shift Estimation for Class-Imbalance Problem: A Bayesian Approach". In: *IEEE/CVF2024*.

Yu, H., Han, J., and Chang, K. (Oct. 2003). „PEBL: Web Page Classification without Negative Examples". In: *IEEE Transactions on Knowledge and Data Engineering* 16.

Zhang, Y., Ju, X., and Tian, Y. (2014). „Nonparallel hyperplane support vector machine for PU learning". In: *2014 10th International Conference on Natural Computation (ICNC)*, pp. 703–708.

Zhao, Y., Xu, Q., Jiang, Y., Wen, P., and Huang, Q. (2022). „Dist-PU: Positive-Unlabeled Learning From a Label Distribution Perspective". In: *Proceedings of the Conference on Computer Vision and Pattern Recognition*. CVPR'22, pp. 14461–14470.

Zhao, Z., Pang, F., Liu, Z., and Ye, C. (2021). „Positive-Unlabeled Learning for Cell Detection in Histopathology Images with&nbsp;Incomplete Annotations”. In: *Medical Image Computing and Computer Assisted Intervention – MICCAI 2021: 24th International Conference, Strasbourg, France, September 27 – October 1, 2021, Proceedings, Part VIII*. event-place: Strasbourg, France. Berlin, Heidelberg: Springer-Verlag, pp. 509–518. ISBN: 978-3-030-87236-6. URL: `https://doi.org/10.1007/978-3-030-87237-3_49`.

Zhou, J., Lu, X., Chang, W., Wan, C., Lu, X., Zhang, C., and Cao, S. (Mar. 2022). „PLUS: Predicting cancer metastasis potential based on positive and unlabeled learning.” In: *PLoS computational biology* 18.3. Place: United States, e1009956. ISSN: 1553-7358 1553-734X.

Zhou, Y., Xu, J., Wu, J., Taghavi, Z., Korpeoglu, E., Achan, K., and He, J. (2021). „PURE: Positive-Unlabeled Recommendation with Generative Adversarial Network”. In: *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. KDD ’21. event-place: Virtual Event, Singapore. New York, NY, USA: Association for Computing Machinery, pp. 2409–2419. ISBN: 978-1-4503-8332-5. URL: `https://doi.org/10.1145/3447548.3467234`.

Appendices

Appendix A

Empirical ELBO derivation

A.1 VaDE empirical ELBO derivation

To obtain the empirical equivalent of the VaDE ELBO, we will introduce two additional properties. First, we prove the following for the Kullback-Leibler divergence between Gaussian distributions:

Theorem A.1. *Kullback-Leibler divergence between two Gaussian distributions $\mathcal{N}(\mu, \Sigma)$ and $\mathcal{N}(m, S)$ is given by:*

$$\begin{aligned} &D_{KL}\left(\mathcal{N}(\mu,\Sigma)||\mathcal{N}(m,S)\right) \\ &\quad= \frac{1}{2}\left(tr(S^{-1}\Sigma) + (m-\mu)^T S^{-1}(m-\mu) - k + \log\frac{det(S)}{det(\Sigma)}\right), \end{aligned} \tag{A.1}$$

where k is the dimensionality of the distribution.

Using Theorem A.1 and the fact that $D_{\mathrm{KL}}(q(z)||p(z)) = \int q(z)\log\frac{q(z)}{p(z)}\,\mathrm{d}z = \int q(z)\log q(z)\,\mathrm{d}z - \int q(z)\log p(z)\,\mathrm{d}z$, one can obtain the following property of the distributions with diagonal covariance matrices:

Theorem A.2. *Let $\Sigma = \mathrm{diag}(\sigma_1^2, \sigma_2^2, ..., \sigma_k^2)$ and $S = \mathrm{diag}(\widetilde{\sigma}_1^2, \widetilde{\sigma}_2^2, ..., \widetilde{\sigma}_k^2)$, and p, q are densities of $\mathcal{N}(\mu, \Sigma)$ and $\mathcal{N}(m, S)$, respectively. Then:*

$$\int q(z)\log p(z)\,\mathrm{d}z = -\frac{1}{2}\sum_{j=1}^{k}\left(\log 2\pi\sigma_j^2 + \frac{\widetilde{\sigma_j}^2}{\sigma_j^2} + \frac{(m_j-\mu_j)^2}{\sigma_j^2}\right). \tag{A.2}$$

We will derive the empirical equivalent of the ELBO using the form of ELBO from Equation (4.22):

$$\mathscr{L}_{\mathrm{ELBO}}(x) = \mathbb{E}_{q(z,c|x)}\left[\underbrace{\log p(x|z)}_{(1)} + \underbrace{\log p(z|c)}_{(2)} + \underbrace{\log p(c)}_{(3)} - \underbrace{\log q(z|x)}_{(4)} - \underbrace{\log q(c|x)}_{(5)}\right] \tag{A.3}$$

We will denote the dimensionality of μ_c, σ_c^2, $\widetilde{\mu}_c$ and $\widetilde{\sigma}_c^2$ as J.

(1) $\mathbb{E}_{q(z,c|x)} \log p(x|z)$. As $z \perp\!\!\!\perp c|x$, we have:

$$\mathbb{E}_{q(z,c|x)} \log p(x|z) = \mathbb{E}_{q(c|x)} \mathbb{E}_{q(z|x)} \log p(x|z) = \mathbb{E}_{q(z|x)} \log p(x|z) \tag{A.4}$$

Using reparametrization trick, we can rewrite this as:

$$\mathbb{E}_{z\sim q(z|x)} \log p(x|z) = \mathbb{E}_{\epsilon\sim\mathcal{N}(0,I)} \log p\left(x|z(\epsilon)\right)$$

and we can replace the expectation with a Monte Carlo estimate with batch of size L:

$$\mathbb{E}_{\epsilon\sim\mathcal{N}(0,I)} \log p(x|z) \approx \frac{1}{L}\sum_{l=1}^{L} \log p\left(x\middle|z(\epsilon^{(l)})\right) \tag{A.5}$$

As $p(x|z) = \mathcal{N}(\mu_x, \sigma_x^2 I)$, we can express this as:

$$\begin{aligned}\frac{1}{L}\sum_{l=1}^{L} \log p\left(x|z\left(\epsilon^{(l)}\right)\right) &= \frac{1}{L}\sum_{l=1}^{L}\sum_{i=1}^{p} \log \frac{1}{\sqrt{2\pi\sigma_{x_i}^2}} \exp\left(-\frac{1}{2}\frac{\left(x_i^{(l)}-\mu_{x_i}\right)^2}{\sigma_{x_i}^2}\right)\\ &= \frac{1}{L}\sum_{l=1}^{L}\sum_{i=1}^{p}\left(-\frac{1}{2}\log 2\pi - \frac{1}{2}\log\sigma_{x_i}^2 - \frac{1}{2}\frac{\left(x_i^{(l)}-\mu_{x_i}\right)^2}{\sigma_{x_i}^2}\right)\\ &= -\frac{1}{2}\left(p\log 2\pi + \sum_{i=1}^{p}\left(\log\sigma_{x_i}^2 + \frac{1}{L}\sum_{l=1}^{L}\frac{\left(x_i^{(l)}-\mu_{x_i}\right)^2}{\sigma_{x_i}^2}\right)\right)\end{aligned} \tag{A.6}$$

(2) $\mathbb{E}_{q(z,c|x)} \log p(z|c)$. Using the mean field assumption, we can express this as:

$$\begin{aligned}\mathbb{E}_{q(z,c|x)} \log p(z|c) &= \int \sum_{c=1}^{K} q(z,c|x)\log p(z|c)\,\mathrm{d}z\\ &= \int \sum_{c=1}^{K} q(z|x)q(c|x)\log p(z|c)\,\mathrm{d}z\\ &= \sum_{c=1}^{K} q(c|x)\int q(z|x)\log p(z|c)\,\mathrm{d}z\end{aligned}$$

Note that $q(z|x) = \mathcal{N}(\widetilde{\mu}, \widetilde{\sigma}^2 I)$ and $p(z|c) = \mathcal{N}(\mu_c, \sigma_c^2 I)$, which allows us to apply Theorem A.2:

$$\begin{aligned}
&\sum_{c=1}^{K} q(c|x) \int q(z|x) \log p(z|c) \, \mathrm{d}z \\
&= \sum_{c=1}^{K} q(c|x) \left(-\frac{1}{2} \sum_{j=1}^{J} \left(\log 2\pi\sigma_{c,j}^2 + \frac{\widetilde{\sigma}_j^2}{\sigma_{c,j}^2} + \frac{(\widetilde{\mu}_j - \mu_{c,j})^2}{\sigma_{c,j}^2} \right) \right) \\
&= -\frac{1}{2} \sum_{c=1}^{K} q(c|x) \left(\sum_{j=1}^{J} \log 2\pi + \sum_{j=1}^{J} \left(\log \sigma_{c,j}^2 + \frac{\widetilde{\sigma}_j^2}{\sigma_{c,j}^2} + \frac{(\widetilde{\mu}_j - \mu_{c,j})^2}{\sigma_{c,j}^2} \right) \right) \\
&= -\frac{1}{2} J \log 2\pi - \frac{1}{2} \sum_{c=1}^{K} q(c|x) \sum_{j=1}^{J} \left(\log \sigma_{c,j}^2 + \frac{\widetilde{\sigma}_j^2}{\sigma_{c,j}^2} + \frac{(\widetilde{\mu}_j - \mu_{c,j})^2}{\sigma_{c,j}^2} \right)
\end{aligned}$$

(3) $\mathbb{E}_{q(z,c|x)} \log p(c)$. As $p(c) = \mathrm{Cat}(c|\boldsymbol{\pi})$ and $z \perp\!\!\!\perp c|x$, we can express this as:

$$\begin{aligned}
\mathbb{E}_{q(z,c|x)} \log p(c) &= \int \sum_{c=1}^{K} q(z, c|x) \log \pi_c \, \mathrm{d}z \\
&= \int \sum_{c=1}^{K} q(z|x) q(c|x) \log \pi_c \, \mathrm{d}z \\
&= \sum_{c=1}^{K} q(c|x) \log \pi_c \int q(z|x) \, \mathrm{d}z \\
&= \sum_{c=1}^{K} q(c|x) \log \pi_c
\end{aligned} \tag{A.7}$$

(4) $\mathbb{E}_{q(z,c|x)} \log q(z|x)$. First using $z \perp\!\!\!\perp c|x$, and then applying Theorem A.2 with $q(z|x) = \mathcal{N}(\widetilde{\mu}, \widetilde{\sigma}^2 I)$, we have:

$$\begin{aligned}
\mathbb{E}_{q(z,c|x)} \log q(z|x) &= \mathbb{E}_{q(c|x)} \mathbb{E}_{q(z|x)} \log q(z|x) \\
&= \mathbb{E}_{q(z|x)} \log q(z|x) \\
&= \int q(z|x) \log q(z|x) \, \mathrm{d}z \\
&= -\frac{1}{2} \sum_{j=1}^{J} \left(\log 2\pi \widetilde{\sigma}_j^2 + \underbrace{\frac{\widetilde{\sigma}_j^2}{\widetilde{\sigma}_j^2}}_{=1} + \underbrace{\frac{(\widetilde{\mu}_j - \widetilde{\mu}_j)^2}{\widetilde{\sigma}_j^2}}_{=0} \right) \qquad \text{(A.8)} \\
&= -\frac{1}{2} \sum_{j=1}^{J} \left(\log 2\pi + \log \widetilde{\sigma}_j^2 + 1 \right) \\
&= -\frac{1}{2} J \log 2\pi - \frac{1}{2} \sum_{j=1}^{J} \left(\log \widetilde{\sigma}_j^2 + 1 \right)
\end{aligned}$$

(5) $\mathbb{E}_{q(z,c|x)} \log q(c|x)$. Reasoning as in point (3):

$$\begin{aligned}
\mathbb{E}_{q(z,c|x)} \log q(c|x) &= \int \sum_{c=1}^{K} q(z,c|x) \log q(c|x) \, \mathrm{d}z \\
&= \sum_{c=1}^{K} q(c|x) \log q(c|x) \int q(z|x) \, \mathrm{d}z \qquad \text{(A.9)} \\
&= \sum_{c=1}^{K} q(c|x) \log q(c|x)
\end{aligned}$$

Denoting $q(c|x)$ as γ_c for simplicity, and putting all the terms above back into the ELBO Equation (A.3), we obtain:

$$\begin{aligned}
\mathscr{L}_{\text{ELBO}}(x) = &-\frac{1}{2}\left(p\log 2\pi + \sum_{i=1}^{p}\left(\log\sigma_{x_i}^2 + \frac{1}{L}\sum_{l=1}^{L}\frac{\left(x_i^{(l)}-\mu_{x_i}\right)^2}{\sigma_{x_i}^2}\right)\right)\\
&-\cancel{\frac{1}{2}J\log}2\pi - \frac{1}{2}\sum_{c=1}^{K}\gamma_c\sum_{j=1}^{J}\left(\log\sigma_{c,j}^2 + \frac{\widetilde{\sigma}_j^2}{\sigma_{c,j}^2} + \frac{(\widetilde{\mu}_j-\mu_{c,j})^2}{\sigma_{c,j}^2}\right)\\
&+\sum_{c=1}^{K}\gamma_c\log\pi_c\\
&+\cancel{\frac{1}{2}J\log}2\pi + \frac{1}{2}\sum_{j=1}^{J}\left(\log\widetilde{\sigma}_j^2+1\right)\\
&-\sum_{c=1}^{K}\gamma_c\log\gamma_c \qquad\qquad (A.10)\\
= &-\frac{1}{2}\left(p\log 2\pi + \sum_{i=1}^{p}\left(\log\sigma_{x_i}^2 + \frac{1}{L}\sum_{l=1}^{L}\frac{\left(x_i^{(l)}-\mu_{x_i}\right)^2}{\sigma_{x_i}^2}\right)\right)\\
&-\frac{1}{2}\sum_{c=1}^{K}\gamma_c\sum_{j=1}^{J}\left(\log\sigma_{c,j}^2 + \frac{\widetilde{\sigma}_j^2}{\sigma_{c,j}^2} + \frac{(\widetilde{\mu}_j-\mu_{c,j})^2}{\sigma_{c,j}^2}\right)\\
&+\frac{1}{2}\sum_{j=1}^{J}\left(\log\widetilde{\sigma}_j^2+1\right)\\
&+\sum_{c=1}^{K}\gamma_c\log\frac{\pi_c}{\gamma_c},
\end{aligned}$$

which is the same as Equation (4.25) in the main text.

A.2 VAE-PU empirical ELBO components

We will derive the empirical equivalent of the VAE-PU ELBO using the form of ELBO from Equation (4.31):

$$\begin{aligned}
&\mathscr{L}_{\text{ELBO}}(x)\\
&=\mathbb{E}_{q(h_y,h_s,c|x,s)}\Bigg[\underbrace{\log p(x|h_y,h_s)}_{(1)}+\underbrace{\log p(h_y|c)}_{(2)}+\underbrace{\log p(s|h_s)}_{(3)}+\underbrace{\log p(h_s)}_{(4)}+\underbrace{\log p(c)}_{(5)} \qquad (A.11)\\
&\qquad\qquad -\underbrace{\log q(h_y|x)}_{(6)}-\underbrace{\log q(h_s|x,s)}_{(7)}-\underbrace{\log q(c|x)}_{(8)}\Bigg]
\end{aligned}$$

The empirical equivalent of the ELBO for VAE-PU model can be derived in a similar way to VaDE. We will denote the dimensionality of μ_c, σ_c^2, $\widetilde{\mu}_c$ and $\widetilde{\sigma}_c^2$ as J_y, and the dimensionality of μ_s, σ_s^2, $\widetilde{\mu}_s$ and $\widetilde{\sigma}_s^2$ as J_s, and number of Monte Carlo examples used for reparametrization trick as L.

$$(1)\quad \mathbb{E}_{q(h_y,h_s,c|x,s)} \log p(x|h_y,h_s) = -\frac{1}{2}\left(p\log 2\pi + \sum_{i=1}^{p}\left(\log\sigma_{x_i}^2 + \frac{1}{L}\sum_{l=1}^{L}\frac{\left(x_i^{(l)} - \mu_{x_i}\right)^2}{\sigma_{x_i}^2}\right)\right)$$

$$(2)\quad \mathbb{E}_{q(h_y,h_s,c|x,s)} \log p(h_y|c) = -\frac{1}{2}J_y\log 2\pi - \frac{1}{2}\sum_{c\in\{0,1\}} q(c|x)\sum_{j=1}^{J_y}\left(\log\sigma_{c,j}^2 + \frac{\sigma_{y,j}^2}{\sigma_{c,j}^2} + \frac{(\mu_{y,j}-\mu_{c,j})^2}{\sigma_{c,j}^2}\right),$$

$$(3)\quad \mathbb{E}_{q(h_y,h_s,c|x,s)} \log p(s|h_s) = -\frac{1}{2}J_s\log 2\pi - \frac{1}{2}\sum_{j=1}^{J_s}\left(\sigma_{s,j}^2 + \mu_{s,j}^2\right),$$

$$(4)\quad \mathbb{E}_{q(h_y,h_s,c|x,s)} \log p(h_s) = s\log\widetilde{s} + (1-s)\log(1-\widetilde{s}),$$

$$(5)\quad \mathbb{E}_{q(h_y,h_s,c|x,s)} \log p(c) = \sum_{c\in\{0,1\}} q(c|x)\log\pi_c,$$

$$(6)\quad \mathbb{E}_{q(h_y,h_s,c|x,s)} \log q(h_y|x) = -\frac{1}{2}J_y\log 2\pi - \frac{1}{2}\sum_{j=1}^{J_y}\left(1 + \log\sigma_{y,j}^2\right)$$

$$(7)\quad \mathbb{E}_{q(h_y,h_s,c|x,s)} \log q(h_s|x,s) = -\frac{1}{2}J_s\log 2\pi - \frac{1}{2}\sum_{j=1}^{J_s}\left(1 + \log\sigma_{s,j}^2\right)$$

$$(8)\quad \mathbb{E}_{q(h_y,h_s,c|x,s)} \log q(c|x) = \sum_{c\in\{0,1\}} q(c|x)\log q(c|x).$$

Plugging all the terms above into the VAE-PU ELBO Equation (A.11), and denoting $q(c|x)$ as γ_c for simplicity, we obtain:

$$
\begin{aligned}
&\mathscr{L}_{\text{ELBO}}(x,s)\\
&= -\frac{1}{2}\left(p\log 2\pi + \sum_{i=1}^{p}\left(\log\sigma_{x_i}^2 + \frac{1}{L}\sum_{l=1}^{L}\frac{\left(x_i^{(l)}-\mu_{x_i}\right)^2}{\sigma_{x_i}^2}\right)\right)\\
&\quad - \cancel{\frac{1}{2}J_y\log 2\pi} - \frac{1}{2}\sum_{c\in\{0,1\}} q(c|x)\sum_{j=1}^{J_y}\left(\log\sigma_{c,j}^2 + \frac{\sigma_{y,j}^2}{\sigma_{c,j}^2} + \frac{(\mu_{y,j}-\mu_{c,j})^2}{\sigma_{c,j}^2}\right)\\
&\quad - \cancel{\frac{1}{2}J_s\log 2\pi} - \frac{1}{2}\sum_{j=1}^{J_s}\left(\sigma_{s,j}^2 + \mu_{s,j}^2\right)\\
&\quad + s\log\widetilde{s} + (1-s)\log(1-\widetilde{s})\\
&\quad + \sum_{c\in\{0,1\}}\gamma_c\log\pi_c\\
&\quad + \cancel{\frac{1}{2}J_y\log 2\pi} + \frac{1}{2}\sum_{j=1}^{J_y}\left(1+\log\sigma_{y,j}^2\right)\\
&\quad + \cancel{\frac{1}{2}J_s\log 2\pi} + \frac{1}{2}\sum_{j=1}^{J_s}\left(1+\log\sigma_{s,j}^2\right)\\
&\quad - \sum_{c\in\{0,1\}}\gamma_c\log\gamma_c\\
&= -\frac{1}{2}\left(p\log 2\pi + \sum_{i=1}^{p}\left(\log\sigma_{x_i}^2 + \frac{1}{L}\sum_{l=1}^{L}\frac{\left(x_i^{(l)}-\mu_{x_i}\right)^2}{\sigma_{x_i}^2}\right)\right)\\
&\quad - \frac{1}{2}\sum_{c\in\{0,1\}}\gamma_c\sum_{j=1}^{J_y}\left(\log\sigma_{c,j}^2 + \frac{\sigma_{y,j}^2}{\sigma_{c,j}^2} + \frac{(\mu_{y,j}-\mu_{c,j})^2}{\sigma_{c,j}^2}\right)\\
&\quad + \frac{1}{2}\sum_{j=1}^{J_y}\left(1+\log\sigma_{y,j}^2\right)\\
&\quad - \frac{1}{2}\sum_{j=1}^{J_s}\left(1+\log\sigma_{s,j}^2 - \sigma_{s,j}^2 - \mu_{s,j}^2\right)\\
&\quad + \sum_{c\in\{0,1\}}\gamma_c\log\frac{\pi_c}{\gamma_c}\\
&\quad + s\log\widetilde{s} + (1-s)\log(1-\widetilde{s})
\end{aligned}
\tag{A.12}
$$

which is the same as Equation (4.32) in the main text. The maximum of ELBO is achieved when $q(c|x)$ itself is obtained by the following formula:

$$q(c|x) = p(c|h_y) = \frac{p(c)p(h_y|c)}{\sum_{c' \in \{0,1\}} p(c')p(h_y|c')} \tag{A.13}$$

Appendix B

Properties of logistic, exponential and inverse sigmoid losses

B.1 Logistic loss

We consider the logistic loss $l_{\log}(t) = \log(1 + e^{-t})$. It is easy to check that it is monotone and convex. In this case, the derivative of conditional risk for an arbitrary value of x can be calculated as follows (note that $\mathbb{E}_{XY} = \mathbb{E}_{Y|X}\,\mathbb{E}_X$):

$$\begin{aligned}
&\frac{\partial}{\partial g(x)}\,\mathbb{E}_{Y|X=x}\, l_{\log}(Y g(X)) \\
&\qquad = \frac{\partial}{\partial g(x)}\,\mathbb{E}_{Y|X=x} \log\left(1 + e^{-Y g(x)}\right) \\
&\qquad = \frac{\partial}{\partial g(x)} P(Y = 1|x) \log\left(1 + e^{-g(x)}\right) + P(Y = -1|x) \log\left(1 + e^{g(x)}\right) \\
&\qquad = P(Y = 1|x) \frac{1}{1 + e^{-g(x)}} \left(-e^{-g(x)}\right) + P(Y = -1|x) \frac{1}{1 + e^{g(x)}} e^{g(x)}
\end{aligned}$$

To obtain a stationary point $g^*_{\log}$, we equate the above derivative to 0 and obtain the following:

$$\begin{aligned}
P(Y=1|x)\frac{1}{1+e^{-g^*_{\log}(x)}}\left(-e^{-g^*_{\log}(x)}\right)+P(Y=-1|x)\frac{1}{1+e^{g^*_{\log}(x)}}e^{g^*_{\log}(x)}&=0\\
P(Y=1|x)\frac{1}{1+e^{-g^*_{\log}(x)}}e^{-g^*_{\log}(x)}&=P(Y=-1|x)\frac{1}{1+e^{g^*_{\log}(x)}}e^{g^*_{\log}(x)}\\
P(Y=1|x)\frac{1+e^{g^*_{\log}(x)}}{e^{-g^*_{\log}(x)}+1}e^{-g^*_{\log}(x)}&=P(Y=-1|x)e^{g^*_{\log}(x)}\\
P(Y=1|x)e^{g^*_{\log}(x)}e^{-g^*_{\log}(x)}&=P(Y=-1|x)e^{g^*_{\log}(x)}\\
P(Y=1|x)&=P(Y=-1|x)e^{g^*_{\log}(x)}\\
e^{g^*_{\log}(x)}&=\frac{P(Y=1|x)}{P(Y=-1|x)}\\
g^*_{\log}(x)&=\log\frac{P(Y=1|x)}{P(Y=-1|x)}
\end{aligned}$$

Note that $g^*_{\log}(x)$ is a nondecreasing function of odds, and the $d(x)$ classification rule based on follows the **Bayesian rule**, classifying the example x to positive class if it's more likely to belong to it over negative class ($P(Y=1|x)\geq P(Y=-1|x)$), as:

$$\begin{gathered}
g^*_{\log}(x)\geq 0\\
\log\frac{P(Y=1|x)}{P(Y=-1|x)}\geq 0\\
\frac{P(Y=1|x)}{P(Y=-1|x)}\geq 1\\
P(Y=1|x)\geq P(Y=-1|x)
\end{gathered}$$

B.2 Exponential loss

We consider the exponential loss $l_{\exp}(t)=e^{-t}$. It is easy to check that it is monotone and convex. Similarly, zero of the derivative of conditional risk for an arbitrary value of x can be calculated as follows:

$$\begin{aligned}
&\frac{\partial}{\partial g(x)}\mathbb{E}_{Y|X=x}\,l_{\exp}(Yg(X))\\
&\quad=\frac{\partial}{\partial g(x)}\mathbb{E}_{Y|X=x}\,e^{-Yg(x)}\\
&\quad=\frac{\partial}{\partial g(x)}P(Y=1|x)e^{-g(x)}+P(Y=-1|x)e^{g(x)}\\
&\quad=P(Y=1|x)-e^{-g(x)}+P(Y=-1|x)e^{g(x)}
\end{aligned}$$

and the optimal classifier is obtained by solving the equation

$$\begin{aligned} P(Y=1|x) - e^{-g^*_{\exp}(x)} + P(Y=-1|x)e^{g^*_{\exp}(x)} &= 0 \\ P(Y=-1|x)e^{g^*_{\exp}(x)} &= P(Y=1|x)e^{-g^*_{\exp}(x)} \\ e^{2g^*_{\exp}(x)} &= \frac{P(Y=1|x)}{P(Y=-1|x)} \\ g^*_{\exp}(x) &= \frac{1}{2}\log\frac{P(Y=1|x)}{P(Y=-1|x)} \end{aligned}$$

B.3 Inverse sigmoid loss

We consider the inverse sigmoid loss $l_{\text{inv}}(t) = (1+e^{-t})^{-1}$. The derivative of conditional risk for a specific value of x can be calculated as follows:

$$\begin{aligned} &\frac{\partial}{\partial g(x)} \mathbb{E}_{Y|X=x}\, l_{\text{inv}}(Y g(X)) \\ &\quad= \frac{\partial}{\partial g(x)} \mathbb{E}_{Y|X=x} \left(1+e^{-Yg(x)}\right)^{-1} \\ &\quad= \frac{\partial}{\partial g(x)} P(Y=1|x)\left(1+e^{-g(x)}\right)^{-1} + P(Y=-1|x)\left(1+e^{g(x)}\right)^{-1} \\ &\quad= P(Y=1|x)\frac{e^{-g(x)}}{(1+e^{-g(x)})^2} + P(Y=-1|x)\frac{e^{g(x)}}{(1+e^{g(x)})^2} \end{aligned}$$

Equating this to 0 yields the equality

$$\frac{P(Y=1|x)}{P(Y=-1|x)} = e^{2g(x)}\left(\frac{1+e^{-g(x)}}{1+e^{g(x)}}\right)^2 == e^{2g(x)}\left(e^{-g(x)}\right)^2 \equiv 1.$$

Thus the stationary point does not exist if $P(Y=1|x) \not\equiv \frac{1}{2}$, and if the condition holds any $g(x)$ is a stationary point (which is obvious, as risk function is equal to 1 in this case).

Appendix C

Example 6.3 – full derivation

Example 6.3. *Let $y(x) = \Phi(x), X \sim \mathcal{N}(0,1)$, and $x \in \mathbb{R}$ (univariate probit model with standard normal predictor), and let propensity score $e_a(x) = \mathbb{I}\{x > a\}$ i.e. above threshold $a \in \mathbb{R}$ all positive observations are labeled. It is easy to check that*

$$
\begin{aligned}
P(Y=1) = \int P(Y=1|X=x) f(x)\,\mathrm{d}x &= \int_{-\infty}^{\infty} \Phi(x)\phi(x)\,\mathrm{d}x \\
&= \int_0^1 z\,\mathrm{d}z = \frac{1}{2}.
\end{aligned}
$$

and Bayes risk of $d_B(x)$ equals

$$
\begin{aligned}
L^* &= \int_{-\infty}^{\infty} \min\big(\Phi(x), 1-\Phi(x)\big)\phi(x)\,\mathrm{d}x \\
&= \int_{-\infty}^{0} \Phi(x)\phi(x)\,\mathrm{d}x + \int_0^{\infty} (1-\Phi(x))\phi(x)\,\mathrm{d}x = \frac{1}{4}.
\end{aligned}
$$

As $s(x) = y(x)\mathbb{I}\{x > a\}$, probability of labeling equals

$$
\begin{aligned}
P(S=1) = \int_{-\infty}^{\infty} P(S=1|X=x) f(x)\,\mathrm{d}x &= \int_{-\infty}^{\infty} s(x)\phi(x)\,\mathrm{d}x \\
&= \int_a^{\infty} \Phi(x)\phi(x)\,\mathrm{d}x = \frac{1}{2}(1-\Phi^2(a))
\end{aligned}
$$

and $P(S=0)=\frac{1}{2}(1+\Phi^2(a))$. *Moreover,* $L^*_{PU}=L^*_{PU}(a)$ *equals for* $a>0$

$$\begin{aligned}
L^*_{PU}(a) &= \mathbb{E}_{X,S=0}\min\big(\widetilde{y}(X,0),1-\widetilde{y}(X,0)\big)\\
&= \left\{\int_{-\infty}^{0}\Phi(x)\phi(x)\,\mathrm{d}x+\int_{0}^{a}(1-\Phi(x))\phi(x)\,\mathrm{d}x\right\}\\
&= \Phi(a)-\frac{\Phi^2(a)}{2}-\frac{1}{4}.
\end{aligned}$$

and analogous calculation for $a\le 0$ *yields* $L^*_{PU}(a)=\Phi^2(a)/2$.

Thus the excess risk of $d_B(x)$ *defined in (6.2) for* $a>0$ *equals*

$$\begin{aligned}
&\mathbb{E}_X\big[\min\big(y(X),1-y(X)\big)\big]\\
&\quad-\mathbb{E}_{X,S}\big[\min\big(\widetilde{y}(X,S),1-\widetilde{y}(X,S)\big)\big]\\
&=\frac{1}{2}-\Phi(a)+\frac{\Phi^2(a)}{2}=\frac{1}{2}\big(\Phi(a)-1\big)^2\ge 0,
\end{aligned}$$

and for $a<0$ *equals* $\frac{1}{4}-\frac{\Phi^2(a)}{2}\ge 0$. *Note that for* $a\to\infty$ *excess risk tends to* 0 *as* $P_{X,S=0}$ *approaches* P_X *in this case and* $d_B^{PU}(x,0)$ *tends to* $d_B(x)$. *For* $a\to-\infty$ *the excess risk tends to 1/4 (risk of* $d_B(x)$*) as the risk of* $d_B^{PU}(x,s)$ *tendto* 0 *0.*

Appendix D

Augmented PU – Balanced accuracy results

Tables D.1 through D.4 correspond to tables 6.2-6.5 in the main text and present U-balanced accuracy for all the performed experiments. This is an important metric, as U-metrics introduce imbalance into the measurements, which might impact accuracy negatively. We can see, however, that the obtained results are close to results presented in the main text – while VP-B+S does not have as overwhelming of an advantage, it still clearly is the best method overall.

Table D.1: U-Balanced accuracy values – Method comparison – Synthetic datasets

c	**Method**	**Synth. 1**	**Synth. 2**	**Synth. 3**	**Synth. SCAR**
0.02	VP+S	**61.29 ± 2.27**	**59.43 ± 2.66**	**60.53 ± 2.30**	**63.52 ± 1.77**
	VP-B+S	61.04 ± 2.33	59.32 ± 2.61	60.40 ± 2.25	63.33 ± 1.73
	LBE+S	49.70 ± 0.35	50.01 ± 0.35	49.93 ± 0.32	49.69 ± 0.33
0.10	VP+S	67.67 ± 0.52	**66.72 ± 0.60**	**67.47 ± 0.61**	**68.49 ± 0.47**
	VP-B+S	**67.71 ± 0.52**	66.45 ± 0.54	67.43 ± 0.58	68.26 ± 0.47
	LBE+S	50.03 ± 0.33	50.83 ± 0.36	50.22 ± 0.32	49.66 ± 0.37
0.30	VP+S	67.49 ± 0.58	65.39 ± 0.59	66.24 ± 0.50	**70.33 ± 0.56**
	VP-B+S	**67.74 ± 0.56**	**65.61 ± 0.63**	**66.35 ± 0.52**	70.17 ± 0.55
	LBE+S	52.86 ± 0.18	52.88 ± 0.33	52.68 ± 0.22	50.00 ± 0.32
0.50	VP+S	**64.11 ± 0.54**	**61.68 ± 0.62**	63.18 ± 0.67	**69.30 ± 0.43**
	VP-B+S	64.09 ± 0.53	61.65 ± 0.61	**63.27 ± 0.70**	69.21 ± 0.44
	LBE+S	55.57 ± 0.49	54.62 ± 0.47	54.96 ± 0.50	56.69 ± 0.44
0.70	VP+S	**59.40 ± 0.78**	58.33 ± 0.73	**58.51 ± 0.65**	65.79 ± 0.56
	VP-B+S	**59.40 ± 0.79**	**58.42 ± 0.69**	58.35 ± 0.60	65.49 ± 0.71
	LBE+S	58.33 ± 0.65	56.61 ± 0.77	56.60 ± 0.66	**67.87 ± 0.59**
0.90	VP+S	52.19 ± 0.57	52.48 ± 0.58	52.33 ± 0.51	59.52 ± 0.98
	VP-B+S	52.19 ± 0.55	52.41 ± 0.57	52.48 ± 0.60	58.64 ± 0.85
	LBE+S	**57.42 ± 0.86**	**56.47 ± 0.92**	**54.63 ± 1.01**	**71.37 ± 1.02**

Table D.2: U-Balanced accuracy values – Method comparison – Real-world datasets

c	**Method**	**MNIST 3v5**	**MNIST OvE**	**CIFAR CT**	**CIFAR MA**	**STL MA**	**CDC-Diabetes**
0.02	VP+S	76.89 ± 1.14	68.32 ± 1.19	87.28 ± 0.49	91.24 ± 0.21	82.13 ± 0.64	50.00 ± 1.48
	VP-B+S	**78.37 ± 1.53**	**74.88 ± 1.51**	**92.46 ± 0.43**	**94.03 ± 0.09**	**83.42 ± 0.69**	**51.16 ± 1.74**
	LBE+S	49.95 ± 0.34	50.14 ± 0.14	50.00 ± 0.40	50.16 ± 0.18	50.17 ± 0.24	49.97 ± 0.20
0.10	VP+S	80.06 ± 0.61	74.69 ± 1.38	91.64 ± 0.32	92.75 ± 0.24	87.19 ± 0.32	57.91 ± 0.78
	VP-B+S	**84.29 ± 0.76**	**83.22 ± 1.23**	**93.46 ± 0.19**	**94.28 ± 0.13**	**88.24 ± 0.30**	**60.93 ± 0.68**
	LBE+S	50.25 ± 0.34	49.79 ± 0.15	50.61 ± 0.38	50.34 ± 0.28	50.78 ± 0.28	49.93 ± 0.19
0.30	VP+S	82.31 ± 0.67	80.74 ± 0.86	93.20 ± 0.26	94.06 ± 0.10	**88.65 ± 0.28**	57.47 ± 0.84
	VP-B+S	**86.95 ± 0.51**	**87.99 ± 0.64**	**94.20 ± 0.16**	**94.78 ± 0.09**	88.32 ± 0.35	**61.60 ± 0.92**
	LBE+S	50.77 ± 0.31	49.79 ± 0.18	55.32 ± 1.30	60.51 ± 4.24	55.90 ± 1.85	49.93 ± 0.17
0.50	VP+S	85.05 ± 0.70	84.64 ± 0.78	94.24 ± 0.24	**94.52 ± 0.12**	**89.34 ± 0.47**	59.37 ± 0.56
	VP-B+S	**88.49 ± 0.83**	**89.86 ± 0.59**	**94.73 ± 0.24**	94.43 ± 0.15	88.60 ± 0.53	**62.98 ± 0.41**
	LBE+S	51.80 ± 0.34	52.90 ± 1.30	72.38 ± 2.72	73.44 ± 2.77	70.34 ± 2.07	62.39 ± 1.83
0.70	VP+S	88.28 ± 0.67	89.25 ± 0.54	94.35 ± 0.32	**94.23 ± 0.17**	**88.63 ± 0.54**	60.07 ± 0.73
	VP-B+S	**90.09 ± 0.64**	**91.30 ± 0.55**	**94.66 ± 0.27**	93.51 ± 0.26	87.97 ± 0.51	62.31 ± 0.62
	LBE+S	54.66 ± 0.65	60.61 ± 1.00	91.25 ± 1.47	90.31 ± 1.01	83.02 ± 1.48	**69.67 ± 1.02**
0.90	VP+S	**87.14 ± 0.74**	**92.30 ± 0.40**	91.97 ± 0.55	92.77 ± 0.45	82.75 ± 1.47	63.01 ± 0.71
	VP-B+S	86.15 ± 0.70	91.84 ± 0.60	92.00 ± 0.55	89.65 ± 0.34	82.43 ± 1.45	58.52 ± 0.53
	LBE+S	67.77 ± 0.98	83.49 ± 0.79	**93.50 ± 0.46**	**93.68 ± 0.46**	**88.35 ± 0.70**	**72.88 ± 0.38**

Table D.3: U-Balanced accuracy values – Decision rule comparison – Synthetic datasets

c	Method	Synth. 1	Synth. 2	Synth. 3	Synth. SCAR
0.02	S-Prophet	73.29 ± 0.35	73.25 ± 0.36	71.37 ± 0.35	73.48 ± 0.35
	Y-Prophet	73.32 ± 0.36	73.25 ± 0.36	71.40 ± 0.36	73.51 ± 0.35
	VP-B	60.94 ± 2.39	**59.42 ± 2.55**	60.14 ± 2.31	**63.33 ± 1.68**
	VP-B+S	**61.04 ± 2.33**	59.32 ± 2.61	**60.40 ± 2.25**	**63.33 ± 1.73**
	VP-B+S + true s(x)	60.81 ± 2.36	59.16 ± 2.68	60.18 ± 2.29	63.23 ± 1.70
	VP-B+S + true y(x)	73.29 ± 0.37	73.22 ± 0.36	71.43 ± 0.36	73.46 ± 0.34
0.10	S-Prophet	72.52 ± 0.31	72.06 ± 0.35	70.49 ± 0.31	73.61 ± 0.35
	Y-Prophet	72.60 ± 0.35	72.13 ± 0.38	70.65 ± 0.37	73.69 ± 0.33
	VP-B	**68.02 ± 0.46**	**66.65 ± 0.57**	**67.62 ± 0.59**	68.16 ± 0.38
	VP-B+S	67.71 ± 0.52	66.45 ± 0.54	67.43 ± 0.58	**68.26 ± 0.47**
	VP-B+S + true s(x)	67.80 ± 0.52	66.57 ± 0.49	67.30 ± 0.62	68.20 ± 0.46
	VP-B+S + true y(x)	72.53 ± 0.35	71.75 ± 0.41	70.42 ± 0.36	73.55 ± 0.33
0.30	S-Prophet	69.91 ± 0.47	69.02 ± 0.52	67.65 ± 0.42	72.69 ± 0.48
	Y-Prophet	70.57 ± 0.43	69.37 ± 0.42	68.49 ± 0.39	73.60 ± 0.35
	VP-B	**68.08 ± 0.52**	**65.85 ± 0.58**	**67.19 ± 0.52**	**70.53 ± 0.49**
	VP-B+S	67.74 ± 0.56	65.61 ± 0.63	66.35 ± 0.52	70.17 ± 0.55
	VP-B+S + true s(x)	67.67 ± 0.57	65.70 ± 0.64	66.48 ± 0.55	70.14 ± 0.51
	VP-B+S + true y(x)	69.30 ± 0.47	68.33 ± 0.52	67.14 ± 0.43	72.61 ± 0.50
0.50	S-Prophet	65.33 ± 0.70	63.89 ± 0.56	63.25 ± 0.71	71.13 ± 0.62
	Y-Prophet	68.13 ± 0.49	66.56 ± 0.48	66.02 ± 0.50	73.18 ± 0.36
	VP-B	**64.95 ± 0.45**	**62.65 ± 0.56**	**64.12 ± 0.70**	**70.31 ± 0.49**
	VP-B+S	64.09 ± 0.53	61.65 ± 0.61	63.27 ± 0.70	69.21 ± 0.44
	VP-B+S + true s(x)	64.08 ± 0.48	61.58 ± 0.60	63.15 ± 0.67	69.30 ± 0.46
	VP-B+S + true y(x)	64.44 ± 0.62	62.68 ± 0.60	62.03 ± 0.82	70.47 ± 0.69
0.70	S-Prophet	57.88 ± 0.51	56.62 ± 0.38	56.71 ± 0.61	66.39 ± 0.48
	Y-Prophet	64.64 ± 0.70	63.34 ± 0.70	62.71 ± 0.73	73.39 ± 0.41
	VP-B	**61.12 ± 0.65**	**59.53 ± 0.68**	**59.84 ± 0.68**	**68.95 ± 0.68**
	VP-B+S	59.40 ± 0.79	58.42 ± 0.69	58.35 ± 0.60	65.49 ± 0.71
	VP-B+S + true s(x)	59.33 ± 0.61	58.45 ± 0.61	58.31 ± 0.70	65.01 ± 0.78
	VP-B+S + true y(x)	57.07 ± 0.73	56.58 ± 0.52	56.44 ± 0.64	65.76 ± 0.39
0.90	S-Prophet	50.38 ± 0.19	50.35 ± 0.21	50.47 ± 0.16	55.62 ± 0.55
	Y-Prophet	61.12 ± 1.16	60.29 ± 1.07	58.82 ± 1.08	72.72 ± 1.00
	VP-B	**58.85 ± 0.62**	**57.12 ± 0.69**	**55.19 ± 0.99**	**71.61 ± 0.91**
	VP-B+S	52.19 ± 0.55	52.41 ± 0.57	52.48 ± 0.60	58.64 ± 0.85
	VP-B+S + true s(x)	52.99 ± 0.40	52.90 ± 0.41	51.86 ± 0.67	60.43 ± 0.93
	VP-B+S + true y(x)	50.94 ± 0.53	51.03 ± 0.41	52.25 ± 0.50	54.80 ± 0.55

Table D.4: U-Balanced accuracy values – Decision rule comparison – Real-world datasets

c	Method	MNIST 3v5	MNIST OvE	CIFAR CT	CIFAR MA	STL MA	CDC-Diabetes
0.02	VP-B	78.32 ± 1.45	74.49 ± 1.49	92.42 ± 0.41	93.91 ± 0.09	**83.46 ± 0.68**	51.12 ± 1.76
	VP-B+S	**78.37 ± 1.53**	**74.88 ± 1.51**	**92.46 ± 0.43**	**94.03 ± 0.09**	83.42 ± 0.69	**51.16 ± 1.74**
0.10	VP-B	84.10 ± 0.65	82.80 ± 1.28	93.35 ± 0.19	**94.28 ± 0.12**	**88.27 ± 0.29**	**61.38 ± 0.77**
	VP-B+S	**84.29 ± 0.76**	**83.22 ± 1.23**	**93.46 ± 0.19**	**94.28 ± 0.13**	88.24 ± 0.30	60.93 ± 0.68
0.30	VP-B	86.75 ± 0.55	**88.04 ± 0.65**	94.12 ± 0.18	**94.85 ± 0.08**	**89.11 ± 0.28**	**63.78 ± 0.78**
	VP-B+S	**86.95 ± 0.51**	87.99 ± 0.64	**94.20 ± 0.16**	94.78 ± 0.09	88.32 ± 0.35	61.60 ± 0.92
0.50	VP-B	88.33 ± 0.88	**90.09 ± 0.57**	94.69 ± 0.25	**94.72 ± 0.10**	**89.81 ± 0.34**	**67.36 ± 0.25**
	VP-B+S	**88.49 ± 0.83**	89.86 ± 0.59	**94.73 ± 0.24**	94.43 ± 0.15	88.60 ± 0.53	62.98 ± 0.41
0.70	VP-B	**90.23 ± 0.66**	**92.15 ± 0.47**	**94.81 ± 0.26**	**94.65 ± 0.15**	**90.82 ± 0.36**	**68.93 ± 0.40**
	VP-B+S	90.09 ± 0.64	91.30 ± 0.55	94.66 ± 0.27	93.51 ± 0.26	87.97 ± 0.51	62.31 ± 0.62
0.90	VP-B	**87.31 ± 0.72**	**93.55 ± 0.45**	**94.07 ± 0.33**	**94.89 ± 0.19**	**90.03 ± 0.73**	**71.38 ± 0.37**
	VP-B+S	86.15 ± 0.70	91.84 ± 0.60	92.00 ± 0.55	89.65 ± 0.34	82.43 ± 1.45	58.52 ± 0.53

Appendix E

Dataset details

PU learning data, due to its nature, presents an unique challenge for benchmarking. Namely, we are unable to accurately evaluate the performance of PU learning methods on real-world datasets, as we do not have access to the true labels of unlabeled examples. In literature, several ways to mitigate this issue are proposed, e.g. custom criterions with properties similar to F1-score, statistical tests or estimations of standard evaluation metrics (Bekker and Davis 2020). However, all of these metrics are rarely reported, less intuitive, and give a limited insight into the performance of the method. A standard alternative approach is to generate PU datasets from existing classification datasets, by randomly selecting a subset of examples to be labeled, and treating the rest as unlabeled. Most importantly, this approach provides the example classes at test time, allowing us to compute standard classification metrics. Another important advantage of this approach is that it allows us to control the label frequency of the datasets, which gives insights into how the method performs under a wide variety of PU learning conditions, including extreme cases of very low label frequency.as well as cases where a big portion of positive examples is already known.

In the main part of the thesis we use four benchmark datasets resulting in 6 different tasks:

- **MNIST**[1] – two different tasks, **3v5** (images of digit *3* are considered positive, *5* – negative) and **OvE** (images of *odd* digits are positive, *even* – negative),

- **CIFAR-10**[2] – two different tasks, **CarTruck** / **CT** (*automobile* images are positive, *truck* – negative) and **MachineAnimal** / **VA** (*airplane*, *automobile*, *ship* and *truck* images are positive, *bird*, *cat*, *deer*, *dog*, *frog* and *horse* – negative),

- **STL-10**[3] – identical classes (but more complex images) as in CIFAR-10, we consider only **MachineAnimal** / **VA** split,

[1] http://yann.lecun.com/exdb/mnist/

[2] https://www.cs.toronto.edu/~kriz/cifar.html

[3] https://cs.stanford.edu/~acoates/stl10/

Table E.1: Benchmark dataset statistics

Name	**Examples**	**Features**	**Class prior π**
MNIST 3v5	13454	784	0.53
MNIST OvE	70000	784	0.51
CIFAR CarTruck	12000	512	0.50
CIFAR MachineAnimal	60000	512	0.40
STL MachineAnimal	13000	512	0.40
Gas Concentrations	4206	129	0.61
CDC Diabetes	148458	38	0.50

- Depending on the chapter:

  (a) **Gas Concentrations**[4] – *Ethanol* examples are positive, *Ammonia* – negative (Chapters 4 and 5),

  (b) **CDC Diabetes**[5] – original classes: positive examples are patients with diabetes, negative examples are healthy patients (Chapters 6 and 7).

Dataset details are listed in the Table E.1. As the research progressed, we decided to replace the Gas Concentration dataset with a more complex CDC Diabetes dataset, aiming to better simulate the real-world PU learning scenario with medical domain data and improved labeling scheme. Note that even though the classes are relatively balanced in base data, labeling process naturally introduces various amounts of label imbalance, controlled by label frequency value c. This in particular results in a strong imbalance for small c; e.g. for CIFAR CarTruck data set and $c = 0.02$ one obtains, on average, 120 labeled examples and 11880 unlabeled ones.

For CIFAR-10 dataset, similarly to the original VAE-PU (Na et al. 2020) approach, pretrained embedding vectors from Information Invariant Clustering (IIC; Ji, Henriques, and Vedaldi (2019)) are used to extract features from images. Pretrained IIC embeddings were used also for STL-10 dataset. Using pretrained embeddings allows us to focus on PU data classification, abstracting the tuning of CNN architectures. As MNIST datasets are significantly simpler, they are treated as tabular datasets together with Gas Concentrations and CDC Diabetes, with only simple preprocessing (feature scaling).

For no-SCAR PU modeling, several labeling schemes were applied in order to construct artificially labeled datasets from those above. In contrast to analysis of VAE-PU in (Na et al. 2020), our experiments consider multiple label frequency values (defined as overall probability of positive examples being labeled $c = P(S = 1 \mid Y = 1)$) and thus instead of constant probability of labeling, a feature-dependent labeling process is used. For MNIST dataset, examples were labeled

[4] https://archive.ics.uci.edu/ml/datasets/gas+sensor+array+drift+dataset

[5] https://archive.ics.uci.edu/dataset/891/cdc+diabetes+health+indicators

according to digit "boldness" – values of each pixel (normalized to (0–1) range) were averaged, and the examples with highest average value were labeled. Number of examples to be labeled is calculated consistently with label frequency, and taking examples with maximum boldness value ensures that the labeled dataset is a biased sample from the positive ($Y = 1$) distribution. CIFAR-10 and STL-10 datasets used "redness" measure defined as $r(x) = (R(x) - G(x)) + (R(x) - B(x))$, where $R(\cdot), G(\cdot), B(\cdot)$ correspond to R, G and B channel pixel values of input image x. Similarly, images with the highest values of $r(x)$ were labeled. "Maximal value" labeling (taking a portion of the dataset with the highest measure values) as opposed to probabilistic sampling (with probabilities based on those measures) aims to obtain a maximally difficult problem – note that in that case, labeled positive examples are maximally different from the unlabeled positive examples according to the labeling metric. While this deviates from the probabilistic, propensity score based labeling assumed by the methods, it also helps to measure robustness of the method against assumption violations. Gas Concentrations dataset used Strategy 1 described by (Gong et al. 2021) – examples were labeled according to their distance from classification boundary, obtained after fitting logistic regression model to the data. Observations located the furthest away from the boundary had the largest probability to be labeled. CDC-Diabetes aims to simulate a more practical, real-world PU scenario – there, labeling (diagnosis) probability scales with age (quadratically) and education level (linearly with subsequent stages of education) to model health awareness increase for senior citizens and more educated people.

We performed experiments for multiple label frequencies ($c \in \{0.02, 0.1, 0.3, 0.5, 0.7, 0.9\}$) in order to account for various PU task difficulties and labeling scenarios. To obtain such datasets, we synthetically generated label vectors S corresponding to each label frequency.