Machine Learning (2020) 109:719–760
https://doi.org/10.1007/s10994-020-05877-5

Learning from positive and unlabeled data: a survey

Jessa Bekker[1] · **Jesse Davis**[1]

Received: 12 November 2018 / Revised: 18 December 2019 / Accepted: 1 March 2020 /
Published online: 2 April 2020

Abstract
Learning from positive and unlabeled data or PU learning is the setting where a learner only has access to positive examples and unlabeled data. The assumption is that the unlabeled data can contain both positive and negative examples. This setting has attracted increasing interest within the machine learning literature as this type of data naturally arises in applications such as medical diagnosis and knowledge base completion. This article provides a survey of the current state of the art in PU learning. It proposes seven key research questions that commonly arise in this field and provides a broad overview of how the field has tried to address them.

Keywords Classification · Weakly supervised learning · PU learning

Mathematics Subject Classification 68T05

1 Introduction

The goal of binary classification is to learn a model that is able to distinguish between positive and negative examples. To do so, an algorithm has access to training data. In the most traditional setting, this data contains both positive and negative examples and is fully labeled, that is, the class value is not missing for any training example. This is among the most widely studied problems in machine learning.

Learning from positive and unlabeled data or PU learning is a variant of this classical set up where the training data consists of positive and unlabeled examples. The assumption is that each unlabeled example could belong to either the positive or negative class. The term PU learning first began to appear in the early 2000s and there has been a surge of interest in this setting in recent years (Liu et al. 2003; Denis et al. 2005; Li and Liu 2005; Elkan and Noto 2008; Mordelet and Vert 2014; Du Plessis et al. 2015a). It fits within the

Editor: Tom Fawcett.

✉ Jessa Bekker
jessa.bekker@kuleuven.be

Jesse Davis
jesse.davis@kuleuven.be

[1] KU Leuven, Leuven, Belgium

long standing interest in developing learning algorithms that do not require fully supervised data, such as learning from positive-only or one-class data (Khan and Madden 2014) and semi-supervised learning (Chapelle et al. 2009). PU learning differs from the former in that it explicitly incorporates unlabeled data into the learning process. It is related to the latter in that it specializes the standard semi-supervised setting, where typically some labeled examples for all classes are available.

One reason that PU learning has attracted attention is that PU data naturally arises in many significant applications. The following are three illustrative examples of applications characterized by PU data. First, personalized advertising uses visited pages and clicks as positive examples of pages and ads of interest. However, all other pages or ads are not necessarily uninteresting and should therefore not be treated as negative examples but as unlabeled ones. Second, medical records usually only list which diseases a patient has been diagnosed with and they usually do not include which diseases a patient does not have. However, the absence of a diagnosis does not mean that a patient does not have a disease. A patient may simply elect not to go to a doctor and moreover many diseases, such as diabetes, often go undiagnosed (Claesen et al. 2015b). Third, consider the task of knowledge base (KB) completion where the goal is to predict which other tuples should belong in an automatically constructed KB. Here, the training data consists of the tuples already in the KB. However, KBs typically only contain facts (i.e., true statements), so there are no negative examples and the truth value of any tuple not in the KB should be considered unknown (Galárraga et al. 2015; Zupanc and Davis 2018).

Motivated by these significant applications, researchers have taken a keen interest in analyzing the PU learning setting. Within PU learning, people have addressed a number of different tasks using a variety of techniques. Despite the breadth, at a high level, the key research questions about PU learning can be formulated rather straightforwardly as:

1. How can we formalize the problem of learning from PU data?
2. What assumptions are typically made about PU data in order to facilitate the design of learning algorithms?
3. Can we estimate the class prior from PU data and why is this useful?
4. How can we learn a model from PU data?
5. How can we evaluate models in a PU setting?
6. When and why does PU data arise in real-world applications?
7. How does PU learning relate to other areas of machine learning?

This survey is structured around giving a comprehensive overview about how the PU learning research community is tackling each of these questions. It concludes with some perspectives about future directions for PU learning research.

2 Preliminaries on PU learning

Learning from positive and unlabeled data (PU learning) is a special case of binary classification. Therefore, we first review binary classification before formally describing the PU learning setting. Then we introduce the labeling mechanism, which is a key concept in PU learning. Finally, we distinguish between two PU learning settings: the single-training-set and case-control scenarios.

Table 1 Labeled training set example

Age	Diabetes family	Fatigue	Pee/day	Blurred vision	y
25	Yes	Yes	7	No	0
63	No	Yes	10	No	1
49	No	No	4	No	0
34	No	Yes	6	Yes	1

The vector of attribute values are the first 5 rows: x = [age, diabetes family, fatigue, pee/day, blurred vision]

Table 2 Positive and Unlabeled training set example for the same dataset as the on in Table 1

Age	Diabetes family	Fatigue	Pee/day	Blurred vision	y	s
25	Yes	Yes	7	No	?	0
63	No	Yes	10	No	1	1
49	No	No	4	No	?	0
34	No	Yes	6	Yes	?	0

2.1 Binary classification

The goal of binary classification is to train a classifier that can distinguish between two classes of instances, based on their attributes. By convention, the two classes are called "positive" and "negative". To train a binary classifier, the machine learning algorithm has access to a set of training examples. Each training example is a tuple (x, y), where x is the vector of attribute values and y is the class value. An example is positive if $y = 1$ and negative if $y = 0$. Traditional learning algorithms work in a supervised setting, where the training data is assumed to be fully labeled. That is, the class value for each training example is observed. Table 1 shows an example of a fully labeled training set. To enable training a correct classifier, the training data is assumed to be an independent and identically distributed (i.i.d.) sample of the real distribution:

$$\begin{aligned} \mathbf{x} &\sim f(x) \\ &\sim \alpha f_+(x) + (1-\alpha) f_-(x), \end{aligned} \tag{1}$$

with class prior $\alpha = \Pr(y = 1)$ and probability density functions of the true distribution f and the positive and negative examples f_+ and f_- respectively.

2.2 PU learning

The goal of PU learning is the same as general binary classification: train a classifier that can distinguish between positive and negative examples based on the attributes. However, during the learning phase, only some of the positive examples in the training data are labeled and none of the negative examples are.

We represent a PU dataset as a set of triplets (x, y, s) with x a vector of attributes, y the class and s a binary variable representing whether the tuple was selected to be labeled. The class y is not observed, but information about it can be derived from the value of s. If the example is labeled $s = 1$, then it belongs to the positive class: $\Pr(y = 1|s = 1) = 1$. When

Table 3 Notation used in this article

Symbol	Description
x	The vector of attributes of an example
$\mathbf{x}$	A set of vectors of attributes of examples
y	Indicator variable for an example to be positive
$\mathbf{y}$	A set of indicator variables for examples to be positive
s	Indicator variable for an example to be labeled
$\mathbf{s}$	A set of indicator variables for examples to be labeled
α	Class prior $\alpha = \Pr(y = 1)$
c	Label frequency $c = \Pr(s = 1 \mid y = 1)$
e	Propensity score function $e(x) = \Pr(s = 1 \mid y = 1, x)$
$f(x)$	Probability density function of the instance space (true population)
$f_+(x)$	Probability density function of the positive instance space
$f_-(x)$	Probability density function of the negative instance space
$f_l(x)$	probability density function of the labeled instance space
$f_u(x)$	Probability density function of the unlabeled instance space
$\hat{\bullet}$	An estimate for $\bullet$

the example is unlabeled $s = 0$, then it can belong to either class. Table 2 gives an example of a positive and unlabeled version of a training set. Table 3 gives an overview of the notation used in this article.

2.3 Labeling mechanism

The labeled positive examples are selected from the complete set of positive examples according to a probabilistic labeling mechanism, where each positive example x has the probability $e(x) = \Pr(s = 1 \mid y = 1, x)$ of being selected to be labeled, called the *propensity score* (Bekker et al. 2019). Hence, the labeled distribution is a biased version of the positive distribution:

$$f_l(x) = \frac{e(x)}{c} f_+(x), \tag{2}$$

with $f_l(x)$ and $f_+(x)$ the probability density functions of the labeled and positive distributions respectively. The normalization constant c is the *label frequency*, which is the fraction of positive examples that are labeled $c = \mathbb{E}_x[e(x)] = \Pr(s = 1 \mid y = 1)$. This can be seen from the following derivation:

$$\begin{aligned} f_l(x) &= \Pr(x \mid s = 1) \\ &= \Pr(x \mid s = 1, y = 1) && \#\textit{by PU definition} \\ &= \frac{\Pr(s = 1 \mid x, y = 1)}{\Pr(s = 1 \mid y = 1)} \Pr(x \mid y = 1) && \#\textit{Bayes' rule} \\ &= \frac{e(x)}{c} f_+(x) \end{aligned}$$

2.4 The single-training-set and case-control scenarios

The positive and unlabeled examples in PU data can originate from two scenarios. Either they come from a single training set, or they come from two independently drawn datasets, one with all positive examples and one with all unlabeled examples. These scenarios are called the single-training-set scenario and the case-control scenario respectively.

The *single-training-set scenario* assumes that the positive and unlabeled data examples come from the same dataset and that this dataset is an i.i.d. sample from the real distribution, like for supervised classification. A fraction c from the positive examples are selected to be labeled, following their individual propensity scores e(x), therefore, the dataset has a fraction αc of labeled examples.

$$\begin{aligned} \mathbf{x} &\sim f(x) \\ &\sim \alpha f_+(x) + (1-\alpha) f_-(x) \\ &\sim \alpha e(x) f_l(x) + (1-\alpha e(x)) f_u(x). \end{aligned} \tag{3}$$

This scenario arises, for example, in personalized advertising, where users only click a subset of the ads of interest. It can also occur in survey data that suffers from under-reporting. That is, sometimes respondents purposely provide incorrect negative responses such as falsely denying that you are a smoker.

The *case-control scenario* assumes that the positive and unlabeled examples come from two independent datasets and that the unlabeled dataset is an i.i.d. sample from the real distribution:

$$\begin{aligned} \mathbf{x}|\mathbf{s}=\mathbf{0} &\sim f_u(x) \\ &\sim f(x) \\ &\sim \alpha f_+(x) + (1-\alpha) f_-(x). \end{aligned} \tag{4}$$

This scenario comes from the setting where two datasets are used and one is known to only have positive examples. For example, when trying to predict one's socioeconomic status from health record, positive examples could be gathered from health centers in upper-class neighborhoods and unlabeled examples from a random selection of health centers.

The observed positive examples are generated from the same distribution in both the single-training-set and case-control scenario. Hence, in both scenarios the learner has access to a set of examples drawn i.i.d. from the true distribution and a set of examples that are drawn from the positive distribution according to the labeling mechanism that is defined by the propensity score $e(x)$. As a result, most methods can handle both scenarios, but the derivation differs. Consequently, one must always consider the scenario when interpreting results and using software.

The single-training-set scenario has received substantially more attention in the literature. Therefore, this survey assumes this scenario. When methods that were originally proposed in a case-control scenario are discussed on a level where this distinction is necessary, we either convert them to the single-training-set scenario or explicitly state that the case-control scenario is assumed.

2.5 Relationship between the class prior and the label frequency

The class prior α and the label frequency c are closely related to each other. Given a PU dataset, if one is known, the expected value of the other can be calculated. The label frequency is defined as the fraction of positive examples that are labeled in all the data:

$$\begin{aligned} c &= \Pr(s = 1 | y = 1) \\ &= \frac{\Pr(s = 1, y = 1)}{\Pr(y = 1)} \\ &= \frac{\Pr(s = 1)}{\Pr(y = 1)}. \qquad \#by\ PU\ definition \end{aligned}$$

The probability $\Pr(s = 1)$ can be counted in the data as the fraction of labeled examples. The probability $\Pr(y = 1)$ is related to the class prior. In the single-training-set scenario, it is equal to the class prior. However, in the case-control scenario, the class prior is defined in the unlabeled data: $\alpha = \Pr(y = 1 | s = 0)$. Here, the probability $\Pr(y = 1)$ is the following:

$$\begin{aligned} \Pr(y = 1) &= \Pr(y = 1 | s = 0) \Pr(s = 0) + \Pr(y = 1 | s = 1) \Pr(s = 1) \\ &= \alpha \Pr(s = 0) + \Pr(s = 1). \end{aligned}$$

To summarize, the conversions between c and α are done as follows:

$$c = \frac{\Pr(s = 1)}{\alpha} \quad \#single\text{-}training\text{-}set\ scenario \tag{5}$$

$$c = \frac{\Pr(s = 1)}{\alpha(1 - \Pr(s = 1)) + \Pr(s = 1)} \quad \#case\text{-}control\ scenario \tag{6}$$

$$\alpha = \frac{1 - c}{c} \frac{\Pr(s = 1)}{1 - \Pr(s = 1)}. \quad \#\ case\text{-}control\ scenario \tag{7}$$

3 Assumptions to enable PU learning

Learning from PU data is not straightforward. There are two possibilities to explain why an example is unlabeled, either:

1. It is truly a negative example; or
2. It is a positive example, but simply was not selected by the labeling mechanism to have its label observed.

Therefore, in order to enable learning with positive and unlabeled data, it is necessary to make assumptions about either the labeling mechanism, the class distributions in the data, or both. The class prior plays an important role in PU learning and many PU learning methods require it as an input. To enable estimating it directly from PU data, additional assumptions need to be made. This section discusses the most commonly made

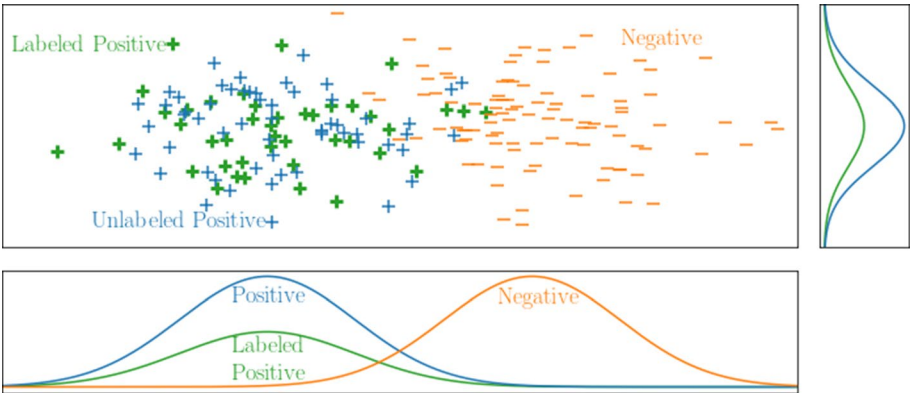

Fig. 1 Example of SCAR PU data. The labeled examples are selected uniformly at random from the positive examples

labeling mechanism and data assumptions to enable PU learning as well as the assumptions made to enable estimating the class prior from PU data.

3.1 Label mechanism assumptions

One approach is to make assumptions about the labeling mechanism. That is, how the examples with an observed positive label were selected.

3.1.1 Selected completely at random

The Selected Completely At Random (SCAR) assumption lies at the basis of most PU learning methods, for example, biased learning methods (Sect. 5.2) and methods that directly incorporate the class prior (Sect. 5.3). It assumes that the set of labeled examples is a uniform subset of the set of positive examples (Elkan and Noto 2008). Figure 1 shows an examples of a PU dataset under the SCAR assumption. This assumption is motivated by the case-control scenario, where it is often reasonable to assume that the labeled dataset is an i.i.d. sample from the positive distribution. However, the SCAR assumption owes its popularity to its ability to reduce PU learning to standard binary classification. This enables applying standard learners to PU problems by either making minor modifications to the data (e.g., weighting it) or the underlying learning algorithm.

Definition 1 (*Selected Completely At Random (SCAR)*) Labeled examples are selected completely at random, independent from their attributes, from the positive distribution. The propensity score $e(x)$, which is the probability for selecting a positive example is constant and equal to the *label frequency* c:

$$e(x) = \Pr(s = 1|x, y = 1) = \Pr(s = 1|y = 1) = c.$$

Under this assumption, the set of labeled examples is an i.i.d. sample from the positive distribution. Indeed, Eq. 2 simplifies to $f_l(x) = f_+(x)$.

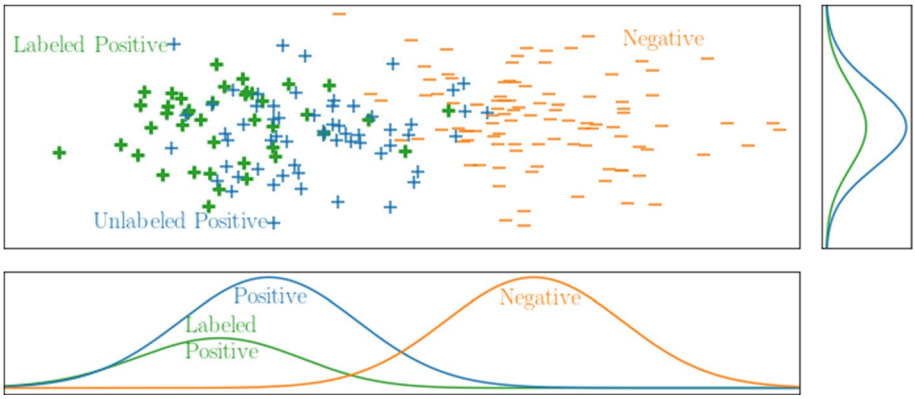

Fig. 2 Example of SAR PU and PGPU data. The labeled examples are a biased sample of the positive examples. The larger the probabilistic gap, the more likely a positive example is selected to be labeled. This means that positive examples which resemble negative examples more, are less likely to be labeled

Under the SCAR assumption, the probability for an example to be labeled is directly proportional to the probability for an example to be positive:

$$\Pr(s = 1|x) = c \Pr(y = 1|x).$$

This enables the use of *non-traditional classifiers*, which are classifiers that predict $\Pr(s = 1|x)$, which are learned by considering the unlabeled examples as negative (Elkan and Noto 2008). These non-traditional classifiers have various interesting properties:

– Non-traditional classifiers preserve the ranking order (Elkan and Noto 2008):

$$\Pr(y = 1|x_1) > \Pr(y = 1|x_2) \Leftrightarrow \Pr(s = 1|x_1) > \Pr(s = 1|x_2).$$

– Training a traditional classifier subject to a desired expected recall, is equivalent to training a non-traditional classifier subject to that recall (Liu et al. 2002; Blanchard et al. 2010)
– Given the label frequency (or class prior), a probabilistic non-traditional classifier can be converted to a traditional classifier, by dividing the outputs by the label frequency $\Pr(y = 1|x) = \Pr(s = 1|x)/c$ (Elkan and Noto 2008).

The SCAR assumption was introduced in analogy with the *Missing Completely A Random assumption (MCAR)* that is common when working with missing data (Rubin 1976; Little and Rubin 2002). However, there is a notable difference between the two assumptions. In MCAR data, the missingness of the variable cannot depend on the value of the variable, where in PU learning this is necessarily the case because all negative labels are missing. The class values are missing completely at random only if just the population of positive examples is considered. Moreno et al. (2012) proposed a new missingness class: *Missing Completely At Random-Class Dependent (MAR-C)*, SCAR belongs to this category.

3.1.2 Selected at random

The Selected At Random (SAR) assumption, is the most general assumption about the labeling mechanism: the probability for selecting positive examples to be labeled depends

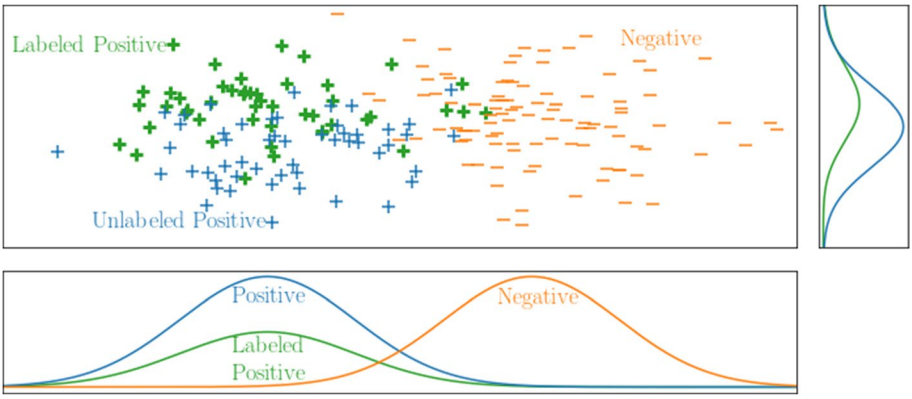

Fig. 3 Example of SAR PU data. The labeled examples are a biased sample of the positive examples. In this case, the labeling mechanism is independent of the probabilistic gap

on its attribute values (Bekker et al. 2019). Figures 2 and 3 show examples of PU datasets under the SAR assumption. This general assumption is motivated by the fact that many PU learning applications suffer from labeling bias. For example, whether someone clicks on a sponsored search ad is influenced by the position in which it is placed. Similarly, whether a patient suffering from a disease will visit a doctor depends on her socioeconomic status and the severity of her symptoms.

Definition 2 (*Selected At Random (SAR)*) Labeled examples are a biased sample from the positive distribution, where the bias completely depends on the attributes and is defined by the *propensity score* $e(x)$:

$$e(x) = \Pr(s = 1|x, y = 1).$$

When the labeling mechanism is understood, incorporating it during the learning phase enables learning an unbiased classifier from SAR PU data. However, when it is not known, additional assumptions are needed to enable learning (Bekker et al. 2019).

3.1.3 Probabilistic gap

Here, it is assumed that positive examples which resemble negative examples more, are less likely to be labeled. The difficulty of labeling is defined by the *probabilistic gap* $\Delta \Pr(x) = \Pr(y = 1|x) - \Pr(y = 0|x)$ (He et al. 2018). The labeling mechanism depends on the attribute values x and is therefore a specific case of SAR, which is illustrated in Fig. 2. This assumption is satisfied naturally in many applications. Diseases with fewer symptoms are more difficult to diagnose, and users are more likely to click on ads that they are more interested in.

Definition 3 (*Probabilistic gap PU (PGPU)*) Labeled examples are a biased sample from the positive distribution, where examples with a smaller probabilistic gap $\Delta \Pr(x)$ are less likely to be labeled. The propensity score is a non-negative, monotone increasing function f of the probabilistic gap $\Delta \Pr(x)$:

$$e(x) = f(\Delta \Pr(x)) = f(\Pr(y = 1|x) - \Pr(y = 0|x)), \quad \frac{d}{dt} f(t) > 0.$$

The observed probabilistic gap $\Delta\tilde{\Pr}(x) = \Pr(s = 1|x) - \Pr(s = 0|x)$ is related to the real probabilistic gap as follows:

$$\Delta\tilde{\Pr}(x) = e(x)(\Delta \Pr(x) + 1) - 1.$$

There are two important properties of this relationship.

1. The observed probabilistic gap is always smaller than or equal to the real probabilistic gap:

$$\Delta\tilde{\Pr}(x) \leq \Delta \Pr(x).$$

Proof

$$\begin{aligned} \Delta\tilde{\Pr}(x) &= e(x)(\Delta \Pr(x) + 1) - 1 \\ &\leq (\Delta \Pr(x) + 1) - 1 \qquad \#\ e(x) \in [0, 1]\ \textit{and}\ \Delta \Pr(x) \geq -1 \\ &= \Delta \Pr(x). \end{aligned}$$

□

From this property it follows that an observed positive probabilistic gap implies a real positive probabilistic gap. This can be used to extract reliable positive examples by selecting examples with an observed positive probabilistic gap (He et al. 2018).

2. Given the probabilistic gap assumption, the observed probabilistic gap maintains the same ordering as the probabilistic gap:

$$\Delta\tilde{\Pr}(x_1) = \Delta\tilde{\Pr}(x_2) \iff \Delta \Pr(x_1) = \Delta \Pr(x_2), \tag{8}$$

$$\Delta\tilde{\Pr}(x_1) > \Delta\tilde{\Pr}(x_2) \iff \Delta \Pr(x_1) > \Delta \Pr(x_2). \tag{9}$$

Proof The equality of 8 is proven by the insight that if two instances have the same probabilistic gaps (i.e., $\Delta \Pr(x_1) = \Delta \Pr(x_2)$), then they must have the same propensity scores, because these are a function of the probabilistic gap $e(x) = f(\Delta(x))$.

$$\begin{aligned} \Delta\tilde{\Pr}(x_1) &= f(\Delta \Pr(x_1))(\Delta \Pr(x_1) + 1) - 1 \\ &= f(\Delta \Pr(x_2))(\Delta \Pr(x_2) + 1) - 1 \\ &= \Delta\tilde{\Pr}(x_2). \end{aligned}$$

The inequality of Eq. 9 is proven by the insight that under the probabilistic gap assumption, an instance with a larger probabilistic gap $\Delta \Pr(x_1) > \Delta \Pr(x_2)$ has a larger propensity

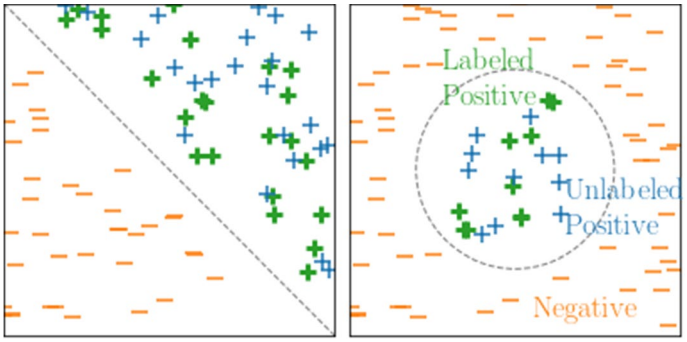

Fig. 4 Examples of separable classes. The first example is linearly separable by a function $f(x_0, x_1) = x_0 + x_1$. The second example is separable by a circle, i.e., by a function $f(x_0, x_1) = -\sqrt{x_0^2 + x_1^2}$

score $e(x_1) = f(\Delta \Pr(x_1)) > f(\Delta \Pr(x_2)) = e(x_2)$ because the propensity score is a monotone increasing function of the probabilistic gap:

$$\begin{aligned} \Delta\tilde{\Pr}(x_1) &= f(\Delta \Pr(x_1))(\Delta \Pr(x_1) + 1) - 1 \\ &> f(\Delta \Pr(x_2))(\Delta \Pr(x_2) + 1) - 1 \\ &= \Delta\tilde{\Pr}(x_2). \end{aligned}$$

□

This property can be used to extract reliable negative examples, by selecting unlabeled examples with an observed probabilistic gap that is smaller than the smallest observed probabilistic gap of the labeled examples (He et al. 2018).

3.2 Data assumptions

The common assumptions about the data distribution are that all unlabeled examples are negative, the classes are separable and the classes have a smooth distribution.

3.2.1 Negativity

The most simple, and most naive, assumption is to assume that the unlabeled examples all belong to the negative class. Despite the fact that this assumption obviously does not hold, it is often used in practice. In the context of knowledge bases, this assumption is commonly referred to as the *closed-world assumption*. The reason why this assumption is popular is because it enables the use of standard machine learning methods for supervised binary classification (Neelakantan et al. 2015). This assumption is simply cited for completeness, and is ignored for the remainder of this survey.

3.2.2 Separability

Under the separability assumption, it is assumed that the two classes of interest are naturally separated. This means that a classifier exists that can perfectly distinguish positive from negative examples. Figure 4 shows some examples of separable classes.

Definition 4 (*Separability*) There exists a function f in the considered hypothesis space that maps all the positive examples to a value that is higher or equal to a threshold τ and all negative examples to a value that is lower than threshold τ:

$$f(x_i) \geq \tau, \quad y_i = 1$$
$$f(x_i) < \tau, \quad y_i = 0.$$

Under this assumption, the optimal classifier can be found by looking for the classifier that classifies all labeled examples as positive and as few as possible examples as negative (Liu et al. 2002; Blanchard et al. 2010). This idea is exploited by the two-step techniques (Sect. 5.1).

3.2.3 Smoothness

According to the smoothness assumption, examples that are close to each other are more likely to have the same label.

Definition 5 (*Smoothness*) If two instances x_1 and x_2 are similar, then the probabilities $\Pr(y = 1|x_1)$ and $\Pr(y = 1|x_2)$ will also be similar.

This assumption allows identifying reliable negative examples as those that are far from all the labeled examples. This can be done by using different similarity (or distance) measures such as tf-idf for text (Li and Liu 2003) or DILCA for categorical attributes (Ienco and Pensa 2016). This assumption is important for two-step techniques (Sect. 5.1). It is also used for graph-based approaches (Pelckmans and Suykens 2009; Yu and Li 2007), local learning (Ke et al. 2017) and to cluster the data into super-instances where all the instances are assumed to have the same label (Li et al. 2009).

3.3 Assumptions for an identifiable class prior

The class prior $\alpha = \Pr(y = 1)$ can be an important tool for PU learning under the SCAR assumption. Therefore, it would be useful if it could be estimated directly from PU data. Unfortunately, this is an ill-defined problem because it is not identifiable: the absence of a label can be explained by either a small prior probability for the positive class or a low label frequency (Scott 2015). In order for the class prior to be identifiable, additional assumption are necessary. This section gives an overview on possible assumptions, listed from strongest to strictly weaker.

1. *Separable Classes/Non-overlapping distributions* Here, the positive and negative distributions are assumed not to overlap (Elkan and Noto 2008; Du Plessis and Sugiyama 2014; Northcutt et al. 2017). The positive examples in the unlabeled data are then all those that are likely to be generated by the same distribution as the labeled examples. When all the unlabeled positive examples are identified, class prior estimation becomes trivial.
2. *Positive subdomain/anchor set* Instead of requiring no overlap between the distributions, it suffices to require a subset of the instance space defined by partial attribute assignment (called the anchor set), to be purely positive (Bekker and Davis 2018a; Liu and Tao 2016; du Plessis et al. 2015b; Scott 2015). The ratio of labeled examples in this

subdomain is equal to the label frequency, while in other parts of the positive distribution, the ratio can be lower.

3. *Positive function/separability* This is a more general version of the positive subdomain assumption, where the subdomain can be defined by any function instead of being limited to partial variable assignments (Ramaswamy et al. 2016). When this assumption was introduced, it was named 'separability', which we find confusing and thus recommend the more intuitive name 'positive function'.
4. *Irreducibility* The negative distribution cannot be a mixture that contains the positive distribution (Blanchard et al. 2010; Jain et al. 2016). All the previous assumption imply irreducibility.

4 PU measures

It is non-obvious how to compute most standard evaluation metrics, such as accuracy, F_1 score, mean square error, etc. from positive and unlabeled data. This introduces challenges both in terms of model evaluation and hyperparameter tuning. The first attempts for addressing this issue focused on proposing metrics that could be computed based on the total number of examples and the number of positive examples. More recent work has explored hypothesis testing and situations where it may be possible to compute standard metrics.

4.1 Metrics for PU data

The most commonly used metric for tuning using PU data is based on the F_1 score, which is defined as:

$$F_1(\hat{\mathbf{y}}) = \frac{2pr}{p+r},$$

with precision $p = \Pr(\mathbf{y} = 1|\hat{\mathbf{y}} = 1)$ and recall $r = \Pr(\hat{\mathbf{y}} = 1|\mathbf{y} = 1)$. Under the SCAR assumption, the recall can be estimated from PU data: $r = \Pr(\hat{\mathbf{y}} = 1|\mathbf{s} = 1)$, however, the precision cannot. The F_1 score cannot be estimated directly from the PU data, but something similar can be. Note that the F_1 score is high when both precision and recall are high. The following performance criterion has the same property and can be estimated from PU data (Lee and Liu 2003):

$$\begin{aligned} \frac{pr}{\Pr(\mathbf{y}=1)} &= \frac{pr^2}{r\Pr(\mathbf{y}=1)} \\ &= \frac{\Pr(\mathbf{y}=1|\hat{\mathbf{y}}=1)r^2}{\Pr(\hat{\mathbf{y}}=1,\mathbf{y}=1)} \\ &= \frac{r^2}{\Pr(\hat{\mathbf{y}}=1)}. \end{aligned} \tag{10}$$

4.2 Hypothesis testing

The G-test is and independence test based on mutual information that can be used for structure learning or feature selection. It turns out that the result of observing independence

with the G-test is the same from supervised and PU data. However, the power of the test differs with a constant correction factor $\frac{1-\alpha}{\alpha}\frac{\Pr(s=0)}{1-\Pr(s=0)}$. Because the correction factor is a constant that depends on the amount of labeled data, one can calculate how much more data is required to get the desired power (Sechidis et al. 2014). The conditional test of independence, which was used for learning the PTAN trees, has similar properties (Calvo et al. 2007; Sechidis and Brown 2015). For feature selection, one is interested in ranking the features in order of mutual information between the features and the label. Interestingly, this order remains the same when the unlabeled examples are considered as negative (Sechidis and Brown 2017).

4.3 Computing standard evaluation metrics

More recently, it has been shown that under certain conditions it is possible to compute (bounds on) traditional metrics used to evaluate learned models (Claesen et al. 2015a; Jain et al. 2017). Effectively, making the SCAR assumption leads to two important insights. First, by estimating the label frequency or class prior, it is possible to compute the expected number of positive examples in the unlabeled data. Second, the rank distributions of the observed positives and the positive examples contained within the unlabeled data should be similar. Combining these two pieces of information enables reasoning about the total number of positive examples (i.e., the sum of the observed positives and the expected number of positives in the unlabeled data) below (above) a given rank. This is precisely the information needed to construct contingency tables, which can be used to derive standard machine learning metrics such as accuracy, the true positive rate, the false positive rate, and precision. Hence, it is possible in this circumstance to report estimates of these metrics.

5 PU learning methods

This section provides an overview of the methods that address PU learning. Most methods can be divided into the following three categories: Two-step techniques, biased learning and class prior incorporation. The two-step technique consists of two steps: (1) identifying reliable negative examples, and (2) learning based on the labeled positives and reliable negatives. Biased learning considers PU data as fully labeled data with class label noise for the negative class. Class prior incorporation modifies standard learning methods by applying the mathematics from the SCAR assumption directly, using the provided class prior. Additionally, methods for learning from relational PU data are discussed.

5.1 Two-step techniques

The two-step technique builds on the assumptions of separability and smoothness. Because of this combination, it is assumed that all the positive examples are similar to the labeled examples and that the negative examples are very different from them. Based on this idea, the two-step technique consists of the following steps (Liu et al. 2003):

Step 1 Identify reliable negative examples. Optionally, additional positive examples can also be generated (Fung et al. 2006).

Table 4 Two-step techniques

Method	Step 1	Step 2	Step 3
S-EM Liu et al. (2002)	Spy	EM NB	ΔE
Roc-SVM Li and Liu (2003)	Rocchio	Iterative SVM	$FNR > 5\%$
Roc-Clu-SVM Li and Liu (2003)	Rocchio*	Iterative SVM	$FNR > 5\%$
PEBL Yu et al. (2002); Yu et al. (2004)	1-DNF	Iterative SVM	Last
A-EM Li and Liu (2005)	Augmented Negatives	EM NB	ΔF
LGN Li et al. (2007)	Single Negative	BN	/
PE_PUC Yu and Li (2007)	PE	(EM) NB	Unspecified
WVC/PSOC Peng et al. (2007)	1-DNF*	Iterative SVM	Vote
CR-SVM Li et al. (2010)	Rocchio*	SVM	/
MCLS Chaudhari and Shevade (2012)	k-means	Iterative LS-SVM	Last
C-CRNE Liu and Peng (2014)	C-CRNE	TFIPNDF	/
Pulce Ienco and Pensa (2016)	DILCA	DILCA-KNN	/
PGPU He et al. (2018)	PGPU	Biased SVM	/

Despite the possibility of choosing the method freely per step, the following combinations where proposed in the literature. Variations of methods are indicated with ∗

Step 2 Use (semi-)supervised learning techniques with the positively labeled examples, reliable negatives, and, optionally, the remaining unlabeled examples.

Step 3 (when applicable) Select the best classifier generated in step 2.

Several methods exist for each one of the steps, which are discussed in the following paragraphs. Despite the possibility of choosing the method freely per step (Liu et al. 2003), most papers propose a fixed combination of methods, which are listed in Table 4.

Step 1: Identifying Reliable Negatives (and Positives) In the first step, unlabeled examples that are very different from the positive examples are selected as reliable negatives. Many methods have been proposed to address this problem. They differ from each other in the way distance is defined and when something is considered as different enough. Many two-step papers addressed text classification problems, therefore, many distance measures originate from that domain (Liu et al. 2002; Li and Liu 2003; Yu et al. 2004; Li and Liu 2005; Fung et al. 2006; Li et al. 2007, 2010; Lu and Bai 2010; Liu and Peng 2014). The following methods have been proposed to identify reliable negative and possibly positive examples:

Spy Some of the labeled examples are turned into spies by adding them to the unlabeled dataset. Then, a Naive Bayes classifier is trained, considering the unlabeled examples as negative, and updated once using expectation maximization. The reliable negative examples are all the unlabeled negative examples for which the posterior probability is lower than the posterior probability of any of the spies (Liu et al. 2002). For this method, it is important to have enough labeled examples, otherwise the set of spies is too small and hence unreliable.

1-DNF First, strong positive features are learned by searching for features that occur more often in the positive data than in the unlabeled data. The reliable negative examples are the examples that do not have any strong positive features (Yu et al. 2004). Because the requirements for positive features are so weak, there might be too many, resulting in very few reliable negative examples. To resolve this, 1-DNFII proposes to discard positive features with an absolute frequency above some threshold (Peng et al. 2007).

Rocchio Based on Rocchio classification, this methods builds a prototype for both the labeled and the unlabeled examples. The prototype is the weighted difference of the mean vector of the tf-idf feature vectors of the objective class and the mean vector of the tf-idf feature vectors of the other class. The unlabeled examples that are closer to the unlabeled prototype than the positive prototype are chosen to be the reliable negatives (Li and Liu 2003). In addition to Rocchio, k-means clustering can be applied to be more selective: every reliable negative that is closer to a positive prototype than a negative one is removed in this step (Li and Liu 2003). Another modification with the aim of being more selective only uses potential unlabeled examples, selected using the cosine similarity, for the negative prototype (Li et al. 2010). Yet another modification is to combine Rocchio with k-means to extract also reliable positive examples in addition to more reliable negatives (Lu and Bai 2010).

PNLH The Positive examples and Negative examples Labeling Heuristic(PNLH) aims to extract both reliable negative and positive examples. First, reliable negatives are extracted using features that more frequently occur in positive data. Subsequently, the sets of reliable positives and negatives are iteratively enlarged by clustering the reliable negatives. Examples that are close to the positive cluster and to no negative cluster are added to the reliable positives. Examples that are close to a negative cluster and not to the positive one are added to the reliable negatives (Fung et al. 2006).

PE Positive Enlargement aims to extract reliable negative and positive examples. A graph-based semi-supervised learning method is used to extract reliable positives and Naive Bayes for reliable negatives (Zhou et al. 2004).

PGPU Under the probabilistic gap assumption (see Sect. 3.1.3), all examples with a positive observed probabilistic gap can confidently be considered as positive, and all examples with an observed probabilistic gap that is smaller than the probabilistic gap of any observed positive example can confidently be considered as negative (He et al. 2018).

k-means All the examples are clustered using k-means. Reliable negative examples are selected from the negative clusters as the furthest ones from the positive examples (Chaudhari and Shevade 2012).

kNN The unlabeled examples are ranked according to their distance to the k nearest positive examples. The unlabeled examples at the greatest distance are selected as reliable negatives (Zhang and Zuo 2009).

C-CRNE Clustering-based method for Collecting Reliable Negative Examples (C-CRNE) is a method that clusters all the examples and takes the clusters without any positive examples as the reliable negatives (Liu and Peng 2014).

DILCA Reliable negatives are selected based on a trainable distance measure DIstance Learning for Categorical Attributes (DILCA), which is designed specifically for categorical attributes (Ienco et al. 2012). This distance measure is learned from the positive examples and then used to detect reliable negatives as the furthest examples.

GPU Generative Positive-Unlabeled (GPU) learns a generative model for the positive distribution, based on the labeled set of positives. The reliable negatives are the unlabeled examples with the lowest probability of being generated by the generative model. The number of reliable negatives is set to be equal to the number of labeled positives (Basile et al. 2018).

Augmented Negatives Instead of selecting reliable negative examples, the unlabeled set is enriched with new examples that are most likely negative. All the unlabeled and added examples are then initialized as negative (Li and Liu 2005). This method is intended for the one-class classification setting where the distribution of negative examples can be different at test time.

Single Negative This method generates a single artificial negative example. This method is intended for an outlier detection setting where very few negative examples are expected in the unlabeled data (Li et al. 2007).

Step 2: (Semi-)Supervised Learning In the second step, the labeled positive examples and reliable negatives are used to train a classifier. Any supervised method, like support vector machines (SVM) or Naive Bayes (NB), can be used for this. Semi-supervised methods, like Expectation Maximization on top of Naive Bayes (EM NB), can also incorporate the remaining unlabeled examples. If semi-supervised methods are used, some methods use the extracted reliable examples from the first step as an initialization that can be changed during the learning process (Liu et al. 2002; Li and Liu 2005; Chaudhari and Shevade 2012), while others fix them and only consider the remaining unlabeled examples for possibly belonging to both classes (Li and Liu 2003; Yu et al. 2004). Apart from existing methods, a few custom methods for PU learning have been proposed:

Iterative SVM In each iteration, an SVM classifier is trained using the positive examples and the reliable negatives. The unlabeled examples that are classified as negative by this classifier are then added to the set of reliable negatives for the next iteration (Yu 2005).

Iterative LS-SVM In each iteration, a non-linear least Squares SVM (LS-SVM) (Suykens and Vandewalle 1999) classifier is trained. During the first iteration, the positive and negative examples come from the initialization. In the later iterations, they come from the classification of the previous iteration. In every iteration, the bias is determined by the desired class ratio (Chaudhari and Shevade 2012).

DILCA-KNN For both the positive and reliable negative examples, a DILCA distance measure is trained (Ienco et al. 2012). For each example, the k nearest positives and k nearest reliable negatives are selected and the average distance to those are calculated with the appropriate distance measure. The class is the one for which it has the lowest average distance (Ienco and Pensa 2016).

TFIPNDF Term Frequency Inverse Positive-Negative Document Frequency is a tf-idf-improved method that weights the terms in documents according to their appearance in positive and negative documents (Liu and Peng 2014).

Step 3 (Optional): Classifier selection Expectation Maximization (EM) generates a new model during every iteration. The local maximum to which EM converges might not be the best model in the sequence. Therefore, different techniques have been proposed to select a model from the sequence:

ΔE The chosen model is the one from the last iteration where the estimated change in the probability of error $\Delta E = \Pr(\hat{y}_i \neq y) - \Pr(\hat{y}_{i-1} \neq y)$ is negative, i.e., the last iteration where the model improved (Liu et al. 2002).

ΔF The chosen model is the one from the last iteration where the estimated change in the F_1 score $\Delta F = F_i/F_{i-1}$ is larger than 1, i.e., the last iteration where the model improved (Li and Liu 2005).

FNR > 5% Stops iterating if more than 5% of the labeled positive examples are classified as negative (Li and Liu 2003).

Vote All the intermediate classifiers are used and their results are combined through weighted voting. The optimal weights can be found through Particle Swarm Optimization (PSO) (Peng et al. 2007).

Last The selected model is the one from the last iteration, when the model has converged or the maximum number of iterations was reached.

5.2 Biased learning

Biased PU learning methods treat the unlabeled examples as negatives examples with class label noise, therefore, this section refers to unlabeled examples as negative. Because the noise for negative examples is a constant, this setting makes the SCAR assumption. The noise is taken into account by, for example, placing higher penalties on misclassified positive examples or tuning hyperparameters based on an evaluation metric that is suitable for PU data. Usually the misclassification penalties or other hyperparameters are chosen through tuning using Eq. 10 (Liu et al. 2003; Claesen et al. 2015d; Zhang et al. 2014; Sellamanickam et al. 2011) or another measure (Shao et al. 2015). Alternatively, they are set based on the true class prior (Hsieh et al. 2015) or so that a balanced classifier is preferred (Mordelet and Vert 2014; Lee and Liu 2003). This approach has been applied to classification, clustering and matrix completion.

5.2.1 Classification

A large fraction of the biased learning methods are based on *support vector machine (SVM)* methods. The original one is biased SVM which is a standard SVM method that

penalizes misclassified positive and negative examples differently (Liu et al. 2003). As an extension, multiple iterations of biased SVM can be executed where misclassified confident unlabeled examples receive an extra penalty (Ke et al. 2012). Weighted unlabeled samples SVM (WUS-SVM) assigns a weight to each unlabeled example, on top of the class penalty, that indicates how likely this examples is to be negative. The weight is the minimum distance to a positive example (Liu et al. 2005).

The noisiness of the negative data makes the learning harder: too much importance might be given to a negative example that is actually positive (Scott and Blanchard 2009). This problem has been addressed by using bagging techniques or using least-square SVMs (LS-SVM) (Suykens and Vandewalle 1999). Bagging SVM learns multiple biased SVM classifiers which are trained on the positive examples and a subset of the negative examples (Mordelet and Vert 2014). Robust Ensemble SVM (RESVM) builds on bagging SVMs by also resampling the positive examples and using a bootstrap approach (Claesen et al. 2015d). Biased least squares SVM (BLSSVM) is a biased version of LS-SVM, which, additionally, enables local learning by using an extra regularization term that favors close-by examples having the same label, using the smoothness assumption (Ke et al. 2017). BLSSVM has been extended to MD-BLSSVM by using the Mahalanobis (Mahalanobis 1936) distance instead of the Euclidean distance (Ke et al. 2018).

RankSVM (RSVM) is an SVM method that minimizes a regularized margin-based pairwise loss (Sellamanickam et al. 2011). In this method, the two classes do not get a different penalty, but the regularization parameter and threshold for classification are set by tuning on Eq. 10. Other hyperplane optimization methods are Biased Twin SVMs (Xu et al. 2014), nonparallel support vector vector machines (NPSVM) (Zhang et al. 2014), and the Laplacian Unit-Hyperplane classifier (LUHC) (Shao et al. 2015).

Weighted *logistic regression* favors correct positive classification over correct negative classification by giving larger weights to positive examples (Lee and Liu 2003). The positive examples are weighted by the negative class prior $\Pr(s = 0)$ and the negative examples by the positive class prior $\Pr(s = 1)$. They show that as a result, the conditional probability that a positive example belongs to the positive class is larger than 0.5 while a negative example will have a conditional probability smaller than 0.5. In principle, a correct classifier would thus be learned. However, when the classes are not separable, the overlapping parts of the instance space might be attributed to the wrong class. This is because the weighting is equivalent to setting the target probability threshold for the non-traditional classifier to $c\Pr(y = 1)$, while it should be $0.5c$ (Elkan 2001). Separable classes can handle this by having 0, 1 probabilities, but non-separable classes are only correctly classified if they are balanced. This is discussed in more detail in Sect. 5.3.2.

5.2.2 Clustering

Topic-Sensitive pLSA (probabilistic latent semantic analysis) is a weighted *constraint clustering* method that introduces must-link constraints between pairs of positive examples and cannot-link constraints between examples from different classes (Zhou et al. 2010). The must-link constraints have stronger weights than the cannot-link constraints. This method is expected to work well when the number of labeled positive examples is small.

5.2.3 Matrix completion

Binary *matrix completion* can also be seen as a PU learning problem: the ones in the matrix are the known positives and the zeros are unlabeled (Hsieh et al. 2015). They assume that in reality, there is a probability matrix of the same size which generated the complete binary matrix. Two binary matrix generation settings are considered: (1) The non-deterministic setting where the complete binary matrix was generated by sampling from the probability matrix, and (2) The deterministic setting where the complete binary matrix was generated by thresholding the probability matrix. The observed matrix is generated by uniform sampling from the complete binary matrix.

In the non-deterministic setting, it is possible to recover the probability matrix, if the true class prior is known. To this end, Shifted Matrix Completion (ShiftMC) minimizes an unbiased estimator for the mean square error loss. This is a special case of the general empirical-risk-minimization based method for incorporating the class prior by preprocessing the data (see Sect. 5.3.2).

In the deterministic setting, the probability matrix cannot be recovered, but the complete binary matrix can. To this end, the matrix factorization method Biased Matrix Completion (BiasMC) penalizes misclassified positives more than misclassified negatives. The penalties are derived from the class prior. Sect. 5.3.2 shows how this is a special case of the rebalancing method for incorporating the class prior by preprocessing the data. An extension to BiasMC for graphs uses the additional information that neighbors are likely similar (Natarajan et al. 2015).

5.3 Incorporation of the class prior

Under the SCAR assumption, the class prior can be used. There are three categories of methods: postprocessing, preprocessing and method modification. Postprocessing trains a non-traditional probabilistic classifier by considering the unlabeled data as negative and modifies the output probabilities, preprocessing changes the dataset by using the class prior, and method modification modifies the methods to incorporate the class prior.

Remember from Sect. 2.5 that knowing the class prior is equivalent to knowing the label frequency c, which is the proportion of labeled positive examples $c = \Pr(s = 1)/\alpha$. The class prior can be determined using methods discussed in Sect. 6 or it can be tuned using evaluation metrics for PU data, which are discussed in Sect. 4.

Under the SAR assumption, in a similar fashion, the propensity score can be incorporated to enable learning. Currently, this has only been explored for the empirical-risk-minimization-based preprocessing method.

5.3.1 Postprocessing

The probability of an example being labeled is directly proportional to the probability of that example being positive, with the label frequency c as the proportionality constant:

$$\Pr(s = 1|x) = c \Pr(y = 1|x).$$

From this result, it follows directly that a non-traditional probabilistic classifier that is trained to predict $\Pr(s = 1|x)$ by considering the unlabeled data as negative can be used to predict the class probabilities $\Pr(y = 1|x) = \frac{1}{c} \Pr(s = 1|x)$ (Elkan and Noto 2008).

Alternatively, when the probabilities are of no importance, the non-traditional classifier can be used directly by changing the target probability threshold τ to $\tau^{PU} = c\tau$. The commonly used $\tau = 0.5$ then results in the decision function $\Pr(s = 1) > 0.5c$. This is equivalent to the decision function $\text{sgn}(\Pr(y = 1|x) - \Pr(y = 0|x)) = \text{sgn}(\frac{2-c}{c}\Pr(s = 1|x) - \Pr(s = 0|x))$ from Zhang and Lee (2005).

5.3.2 Preprocessing

The goal of preprocessing, is to create a new dataset from a PU dataset, which can be used by methods that expect fully supervised data to train the best possible model for the PU data. The proposed methods can be ordered into three categories: rebalancing methods, methods that incorporate the label probabilities and, empirical-risk-minimization-based methods.

Rebalancing Methods As seen before, a non-traditional classifier, trained on the positive and unlabeled data, gives the same classification as a traditional classifier, if the target probability threshold τ is set appropriately. Instead of changing the threshold, the rebalancing method from Elkan (2001) can be employed to weight the data so that the classifier trained on the weighted data will give the same classification with the same target probability threshold as the traditional classifier. Given the target probability threshold for the traditional classifier τ, the target probability threshold for the non-traditional classifier would be $\tau^{PU} = c\tau$. To move the target probability from τ to τ^{PU} in the non-traditional classifier, the data needs to be weighted as follows:

$$\begin{aligned} w^+ &= \tau(1 - \tau^{PU}) & w^- &= (1 - \tau)\tau^{PU} \\ &= \tau(1 - c\tau) & &= (1 - \tau)c\tau \\ &= (1 - c\tau) & &= (1 - \tau)c, \end{aligned}$$

where w^+ and w^- are the weights for positive and negative examples respectively. In the last step, both weights were divided by τ to simplify the formula as this does not affect the learning result. When the target probability is $\tau = 0.5$, this reduces to

$$w^+ = 1 - c/2 \quad w^- = c/2,$$

which is equivalent to the result used for BiasMC (Hsieh et al. 2015). If the true class prior is $\alpha = 0.5$, the result reduces to

$$\begin{aligned} w^+ &= 1 - c\alpha & w^- &= c\alpha \\ w^+ &= Pr(s = 0) & w^- &= \Pr(s = 1) \end{aligned}$$

which are the weights used for weighted logistic regression (Lee and Liu 2003).

Rank Pruning was proposed to be more robust to noise. To this end, it first cleans the data based on the class prior and the expected positive label noise (both of which are estimated in a first phase, see Sect. 6), with the goal of only keeping confident positive and negative examples. The confident examples are then weighted to get the correct class prior (Northcutt et al. 2017).

Rebalancing methods are only appropriate when one is interested in classification on the given target threshold τ, but not for returning the unbiased estimates of the probability $\Pr(y = 1|x)$.

Incorporation of the Label Probabilities Elkan and Noto (2008) proposed to duplicate the unlabeled examples to let them count partially as positive and partially as negative. The weights are the probabilities of the unlabeled examples being positive and negative respectively. The labeled examples are certain to be positive and are therefore added as positive examples with weight 1. The probability for an unlabeled example to be positive is

$$\Pr(y = 1|s = 0, x) = \frac{1-c}{c} \frac{\Pr(s = 1|x)}{1 - \Pr(s = 1|x)}.$$

To generate the weighted dataset like this, first a non-traditional classifier to predict $\Pr(s = 1|x)$ needs to be trained.

Empirical-Risk-Minimization Based Methods The goal of preprocessing the PU data is that the classifier learned from the resulting dataset is expected to be equal to the classifier trained from a fully labeled dataset. In an empirical risk minimization framework, this means finding the classifier g that minimizes the risk, given some loss function L

$$R(g) = \alpha \mathbb{E}_{f_+}\left[L^+(g(x))\right] + (1-\alpha)\mathbb{E}_{f_-}\left[L^-(g(x))\right],$$

where $L^+(\hat{y})$ and $L^-(\hat{y})$ are the losses for positive and negative examples respectively. The following are some popular loss functions:

$$\begin{aligned}
\text{MAE}: \quad & L^+(\hat{y}) = 1 - \hat{y} \quad & L^-(\hat{y}) = \hat{y}, \\
\text{MSE}: \quad & L^+(\hat{y}) = (1-\hat{y})^2 \quad & L^-(\hat{y}) = \hat{y}^2, \\
\text{Log Loss}: \quad & L^+(\hat{y}) = -\ln \hat{y} \quad & L^-(\hat{y}) = -\ln(1-\hat{y}).
\end{aligned}$$

Empirical-Risk-Minimization based-methods, such as SVMs, logistic regression and deep networks, minimize the empirical risk, which is calculated from the data as follows:

$$\begin{aligned}
\hat{R}(g|\mathbf{x}, \mathbf{y}) &= \alpha \frac{1}{|\mathbf{y} = \mathbf{1}|} \sum_{x:\mathbf{x}|\mathbf{y}=\mathbf{1}} L^+(g(x)) + (1-\alpha)\frac{1}{|\mathbf{y} = \mathbf{0}|} \sum_{x:\mathbf{x}|\mathbf{y}=\mathbf{0}} L^-(g(x)) \\
&= \frac{1}{|\mathbf{y}|}\left(\sum_{x:\mathbf{x}|\mathbf{y}=\mathbf{1}} L^+(g(x)) + \sum_{x:\mathbf{x}|\mathbf{y}=\mathbf{0}} L^-(g(x)) \right).
\end{aligned} \tag{11}$$

In PU data, the empirical risk cannot be calculated directly because not all the class values are observed. However, the PU data and the labeling mechanism can be used to create a new, weighted dataset that is expected to give the same empirical risk as the fully labeled data. Next, the risk is rewritten in terms of expectations over the labeled and unlabeled distributions. Then, it is shown how to create the data which gives the same empirical risk when using the standard formula 11 which is used by standard methods and implementations.

The expectation over the negative distribution can be formulated in terms of expectations over the general and the positive distributions, using Eq. 1. The expectation over the positive distribution can be formulated in terms of an expectation over the labeled distribution and the propensity score, using Eq. 2:

$$
\begin{aligned}
R(g) &= \alpha \mathbb{E}_{f_+}\left[L^+(g(x))\right] + (1-\alpha)\mathbb{E}_{f_-}\left[L^-(g(x))\right] \\
&= \alpha \mathbb{E}_{f_+}\left[L^+(g(x))\right] + \mathbb{E}_{f}\left[L^-(g(x))\right] - \alpha \mathbb{E}_{f_+}\left[L^-(g(x))\right] \\
&= \alpha \mathbb{E}_{f_+}\left[L^+(g(x)) - L^-(g(x))\right] + \mathbb{E}_{f}\left[L^-(g(x))\right] \\
&= \alpha \mathbb{E}_{f_l}\left[\frac{c}{e(x)}\left(L^+(g(x)) - L^-(g(x))\right)\right] + \mathbb{E}_{f}\left[L^-(g(x))\right].
\end{aligned}
$$

In the case-control scenario, the expectation over the general distribution can simply be replaced by the expectation over the unlabeled distribution. Therefore, the empirical risk is calculated as follows:

$$
\begin{aligned}
\hat{R}(g|\mathbf{x},\mathbf{s}) = &\frac{\alpha}{|\mathbf{s}=\mathbf{1}|}\sum_{x:\mathbf{x}|\mathbf{s}=\mathbf{1}}\left(\frac{c}{e(x)}\left(L^+(g(x)) - L^-(g(x))\right)\right) \\
&+ \frac{1}{|\mathbf{s}=\mathbf{0}|}\sum_{x:\mathbf{x}|\mathbf{s}=\mathbf{0}} L^-(g(x)). \qquad \#\ \textit{case-control}
\end{aligned}
$$

Hence, the new dataset is created by adding all unlabeled examples as negative with weight $\frac{1}{|\mathbf{s}=\mathbf{0}|}$, and all labeled examples both as positive with weight $\frac{1}{|\mathbf{s}=\mathbf{1}|}\frac{\alpha c}{e(x)}$ and as negative with weight $-\frac{1}{|\mathbf{s}=\mathbf{1}|}\frac{\alpha c}{e(x)}$.

For the single-training-test scenario, the general distribution is a combination of the labeled and unlabeled distributions (Eq. 3), which reduces the risk to:

$$
\begin{aligned}
R(g) = &\alpha c \mathbb{E}_{f_l}\left[\frac{1}{e(x)}L^+(g(x)) + \left(1 - \frac{1}{e(x)}\right)L^-(g(x))\right] \\
&+ (1-\alpha c)\mathbb{E}_{f_u}\left[L^-(g(x))\right]. \qquad \#\ \textit{single-training-set}
\end{aligned}
$$

And the empirical risk to:

$$
\begin{aligned}
\hat{R}(g|\mathbf{x},\mathbf{s}) = &\frac{\alpha c}{|\mathbf{s}=\mathbf{1}|}\sum_{x:\mathbf{x}|\mathbf{s}=\mathbf{1}}\left(\frac{1}{e(x)}L^+(g(x)) + \left(1 - \frac{1}{e(x)}\right)L^-(g(x))\right) \\
&+ \frac{1-\alpha c}{|\mathbf{s}=\mathbf{0}|}\sum_{x:\mathbf{x}|\mathbf{s}=\mathbf{0}}(L^-(g(x))) \\
= &\frac{1}{|\mathbf{s}|}\left(\sum_{x:\mathbf{x}|\mathbf{s}=\mathbf{1}}\left(\frac{1}{e(x)}L^+(g(x)) + \left(1 - \frac{1}{e(x)}\right)L^-(g(x))\right)\right. \\
&\left.+ \sum_{x:\mathbf{x}|\mathbf{s}=\mathbf{0}}(L^-(g(x)))\right). \qquad \#\ \textit{single-training-set}
\end{aligned}
$$

Hence, the new dataset is created by adding all unlabeled examples as negative with weight 1 and all labeled examples both as positive with weight $\frac{1}{e(x)}$ and as negative with weight $(1 - \frac{1}{e(x)})$.

This general weighting method was proposed in the single-training-set scenario as the first SAR PU learning method (Bekker et al. 2019) but it already existed before under the SCAR assumption (Steinberg and Scott Cardell 1992; Du Plessis et al. 2015a; Kiryo et al. 2017). The ShiftMC method for matrix completion is also a special case of this method under the SCAR assumption, using the MSE loss (Hsieh et al. 2015).

du Plessis et al. (2014) proposed another risk estimator, which simply reweights the examples and does not introduce duplicates (du Plessis et al. 2014). However,the derivation is limited to 0–1 predictions and the method is biased, unless the loss functions sum to one $L^+(\hat{y}) + L^-(\hat{y}) = 1$, which can only be achieved with non-convex functions.

5.3.3 Method modification

Many machine learning methods are based on counts of positive and negative examples in subsets of the data. The counts are used to calculate (conditional) probabilities, support, coverage or other metrics that are used to make decisions or set parameters. The counts can be estimated using the same rationale as were used for data weighting (Elkan and Noto 2008).

The PU tree learning algorithm POSC4.5, one of the first PU learning methods, needs the count of positive and negative examples in every considered split for the three. They estimate the number of positives in node i as $\hat{P}_i = \min\{\frac{1}{c}L_i, T_i\}$ and the negatives as $\hat{N}_i = T_i - \hat{P}_i$, where L_i and T_i are the number of labeled and total examples in that node (Denis et al. 2005). This corresponds to empirical-risk-minimization-based weighing.

Ward et al. (2009) proposed an expectation maximization method on top of logistic regression. The expectation step finds the expected class labels and the maximization step trains the logistic regression model using the expected class labels, followed by rebalancing the model according using the class prior.

For Naive Bayes methods, the probabilities $\Pr(x^{(i)}|y)$, with $x^{(i)}$ the ith attribute of x, are key. For $y = 1$, these can be directly estimated from the labeled data as

$$\Pr(x^{(i)}|y = 1) = \Pr(x^{(i)}|s = 1), \tag{12}$$

and for $y = 0$ these can be calculated, somewhat less straightforwardly, as follows:

$$\Pr(x^{(i)}|y = 0) = \frac{\Pr(x^{(i)}) - \alpha \Pr(x^{(i)}|y = 1)}{1 - \alpha}. \tag{13}$$

This insight was used to develop PNB, the first Naive Bayes algorithm for PU learning (Denis et al. 2003). It was originally proposed for document classification, but was later generalized to general discrete attributes and incorporate the of Laplace correction (Calvo et al. 2007). In that same paper an averaging method is presented that can incorporate a distribution over the class prior instead of an exact value. Positive Tree Augmented Naive Bayes (PTAN) builds further on PNB, but also needs to calculate the conditional mutual information between variables i and k for structure learning:

$$\sum_j \sum_l \Pr(x^{(i)} = j, x^{(k)} = l, y = 1) \log \frac{\Pr(x^{(i)} = j, x^{(k)} = l|y = 1)}{\Pr(x^{(i)} = j|y = 1)\Pr(x^{(k)} = l|y = 1)}$$
$$+ \Pr(x^{(i)} = j, x^{(k)} = l, y = 0) \log \frac{\Pr(x^{(i)} = j, x^{(k)} = l|y = 0)}{\Pr(x^{(i)} = j|y = 0)\Pr(x^{(k)} = l|y = 0)},$$

all these probabilities can be calculated by using Eqs. 12, 13, and:

$$\Pr(x^{(i)} = j, x^{(k)} = l, y = 1) = \alpha \Pr(x^{(i)} = j, x^{(k)} = l|s = 1)$$
$$\Pr(x^{(i)} = j, x^{(k)} = l, y = 0) = (1 - \alpha)\Pr(x^{(i)} = j, x^{(k)} = l|y = 0).$$

Similarly, PU learning methods have been proposed for other Bayesian classifiers. Averaged One-Dependence Estimator (AODE) (Webb et al. 2005) has been extended to

PAODE, Hidden Naive Bayes (HNB) (Jiang et al. 2009) to PHNB, and Full Bayesian network Classifier (FBC) (Su and Zhang 2006) to PFBC (He et al. 2011). Some of these methods were further extended to uncertain Bayesian methods, where the attribute values are uncertain: UPNB (He et al. 2010) and UPTAN (Gan et al. 2017), where this last method uses Uncertain Conditional Mutual Information (UCMI) for structure learning (Liang et al. 2012).

5.4 Relational approaches

A common task for relational data is to complete automatically constructed knowledge bases or networks by finding new relationships. This task can be seen as PU learning, because everything that is already in the knowledge base or network is known to be true and everything that can possibly be added is unlabeled. Most methods make the *closed-world* assumption and learn models by assuming everything that is not in the knowledge base is negative. However, a few methods have been proposed that do make the *open-world* assumption, which makes it explicit that the data is incomplete.

When the SCAR assumption holds in the relational PU data, then, relational versions of classic class prior incorporation methods can be used to enable learning (Bekker and Davis 2018b). TI*c*ER, a relational version of TI*c*E (Sect. 6.3) can estimate the class prior directly from the relational PU data.

The PosOnly setting of the relational rule learning system Aleph (Srinivasan 2001) makes the separability assumption and looks for the simplest theory that covers all positive examples and introduces as few new facts as possible (Muggleton 1996).

RelOCC is a relational one-class classification method which, based on the smoothness assumption, introduces a tree-based distance method (Khot et al. 2014). They do not use unlabeled examples at training time, so, although related, it is not truly PU learning.

The AMIE+ rule learning system for knowledge base completion introduces the partial completeness assumption. It assumes that if for a subject and relationship at least one object is known, then all objects for this subject and relationship are known. For example, if `taughtby(bigdata,jesse)`, then it is assumed that the knowledge base contains all Jesse's classes. Using the partial completeness assumption, the confidence of potential rules can be estimated more precisely (Galárraga et al. 2015). The RC confidence score makes an even more precise estimate, by making a rule-specific SCAR assumption and taking the expected relation cardinalities, i.e., the number of objects/subjects per subject/object and rule combination, into account (Zupanc and Davis 2018).

PULSE, a relational PU learning algorithm for disjunctive concepts was proposed in the context of relational grounded language learning (Blockeel 2017). In their setting, the positive class can have a limited number of k subclasses. They assume that for each subclass, the SCAR assumption holds, but do not necessary have the same label frequencies.

5.5 Other methods

For completeness, this section lists PU methods that do not fit in any of the considered categories.

Generative Adversarial Networks (GANs) have recently been introduced for PU learning, where they can model the positive and negative distributions (Hou et al. 2018; Chiaroni et al. 2018).

Co-training is a semi-supervised learning technique that learns two models, based on two views of the data, where the goal is to find two models that agree (Blum and Mitchell 1998). This idea has been applied to PU learning as well (Denis et al. 2003; Zhou et al. 2012).
Data stream classification with PU data has been addressed by multiple works (Li et al. 2009; Nguyen et al. 2011; Qin et al. 2012; Liang et al. 2012; Chang et al. 2016).
Expectation Maximization (EM) can be used for SAR PU data with the additional assumption that the propensity scores only depend on a known subset of the attributes. An EM approach is then used to simultaneously train the classifier and a model for estimating the propensity scores (Bekker et al. 2019).

5.6 Comparison of PU learning methods

The primary consideration for choosing a PU learning method is to ascertain which assumptions are mostly likely to hold for the application at hand. If separability holds, then this would favor the use of two-step techniques. If SCAR holds, then one would use biased learning or methods that incorporate the class prior. If both separability and SCAR hold, the choice depends on how clearly separated the two classes are. If the classes are separable, but very close to each other, separating the two classes correctly is hard for two-step techniques, so exploiting SCAR is likely more effective. However, if the classes are very clearly separated, the two-step techniques are favored, because, given a clear separation, they are more robust against deviations from the SCAR assumption. Currently, not many methods exist that are tailored towards the SAR and PGPU assumptions. Currently, the only PGPU method is a two-step technique that also assumes separability (He et al. 2018). Note that this method is preferred to other two-step techniques, because it builds on the PGPU assumption to find the decision boundary.

If one is interested in unbiased estimates of the true probabilities $\Pr(y = 1|x)$ under the SCAR or SAR assumption, then empirical-risk estimation methods should be considered: ERM data reweigthing (Sect. 5.3.2), ShiftMC (Hsieh et al. 2015), or POS4.5 (Denis et al. 2005). The downside of ERM data reweighting is the use of negative weights, which not all classifiers and implementations can handle. The Naive Bayes method PNB (Denis et al. 2003) and it extensions also output unbiased probabilities. Rebalancing the data (Sect. 5.3.2), or rebalancing/penalizing the classes in biased learning (Sect. 5.2) are not suited for unbiased probabilities, but are expected to find the correct decision boundary.

Rebalancing and class prior incorporation methods are sensitive to the SCAR assumption. Ensemble methods provide more robustness (Claesen et al. 2015d; Mordelet and Vert 2014). Alternatively, the smoothness assumption can be leveraged to relax the SCAR assumption (Ke et al. 2017; Ke et al. 2012; Liu et al. 2005; Sellamanickam et al. 2011).

6 Class prior estimation from PU data

Knowledge of the class prior significantly simplifies PU learning under the SCAR assumption. Therefore, it is very useful to estimate it from PU data directly. To this end, a number of methods have been proposed.

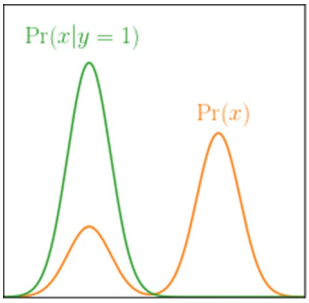

Fig. 5 Partial matching. The goal of partial matching is to find the class prior α that minimizes the divergence between the scaled distributions. This figure is based on Figure 1 in Du Plessis and Sugiyama (2014)

6.1 Non-traditional classifier

When the classes are separable, in principle a non-traditional classifier $g(x)$ that predicts $\Pr(s = 1|x)$ can be trained that maps all negative examples to 0 and all positive examples to $\Pr(s = 1|y = 1) = c$. Based on this insight, Elkan and Noto (Elkan and Noto 2008) suggest to train a classifier on part of the data while keeping a separate validation set. Then, they estimate the label frequency as the average predicted probability of a labeled validation set example (Elkan and Noto 2008). This method requires well-calibrated probabilistic classifiers. Methods such as Platt scaling (Platt 1999), isotonic regression (Zadrozny and Elkan 2002) or beta calibration (Kull et al. 2017) can be used to calibrate classifiers that do not output well-calibrated probabilities. Rank pruning is a more robust method based on a non-traditional classifier g that is based on confident examples: an example x is confidently positive when $g(x) \geq \Pr(\hat{s} = 1|s = 1)$, with $\hat{s}$ the classification by g (Northcutt et al. 2017). The label frequency is calculated from the labeled and unlabeled confident positive examples. This estimation is expected to be correct, as long as the confident positive examples contain no negative examples. Therefore, the method is more robust with regard to the calibration of g and class overlap in the low probability regions. Additionally, rank pruning can handle negative examples that are wrongly labeled in a similar way.

Another method based on a non-traditional classifier uses the insight that the probability $\Pr(s = 1|x) = c \Pr(y = 1|x)$, which is estimated by $g(x)$, is equal to the label frequency c when the true conditional class probability is $\Pr(y = 1|x) = 1$ (Liu and Tao 2016). Under the positive subdomain assumption, there will be instances x for which $\Pr(y = 1|x) = 1$ and hence, the label frequency can be estimated as $c = \max g(x)$.

6.2 Partial matching

The partial matching approach assumes non-overlapping classes. It uses a density estimation method to estimate the positive distribution, based on the labeled examples, and the complete distribution, based on all the data (Du Plessis and Sugiyama 2014). The class prior is found by minimizing the difference between the scaled positive distribution, where the scale factor is the class prior. The method is illustrated in Fig. 5.

The partial matching approach does not work well when the positive and negative distribution overlap. In this case, the correct class prior would give a large divergence in the regions with overlap. By minimizing the divergence, these regions will favor an overestimate of the class prior. To relax the non-overlapping distributions assumption to the

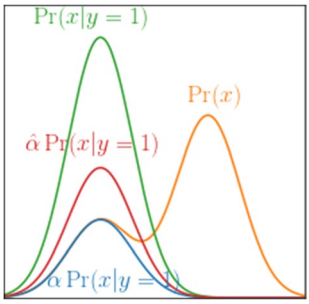

Fig. 6 Partial matching with overlap. When the classes overlap, the original partial mapping method would result in an overestimate for alpha $\hat{\alpha} > \alpha$, like the red line. Using a penalized divergence makes sure that the α -scaled positive distribution does not surpass the total distribution

positive subdomain assumption, penalized divergences were introduced (du Plessis et al. 2015b). These give higher penalties to class priors that result in $\alpha \Pr(x|y = 1) > \Pr(x)$ for some x. Intuitively, this finds the class prior that scales the positive distribution as closely to the total distribution, without ever surpassing it. The method is illustrated in Fig. 6

6.3 Decision tree induction

Tree Induction for c Estimation (TIcE) estimates the label frequency c under the positive subdomain assumption (Bekker and Davis 2018a). It makes the observation that the label frequency remains the same when considering a subdomain of the data and that the fraction of labeled examples in that subdomain provides a natural lower bound on the label frequency. Using a decision tree induction method, it searches for the subdomain that implies the largest lower bound and returns that as the label frequency estimate. Under the positive subdomain assumption, this lower bound is indeed expected to be the label frequency. This method is closely related to the last non-traditional classifier method (Liu and Tao 2016) but differs in that it is more robust and faster. It is more robust because it takes the maximum over sets of instances (subdomains) as opposed to single instances. It is faster because it does not need to train a full tree and instead concentrates on the branches that can give a stricter lower bound.

6.4 Receiver operating characteristic (ROC) approaches

In the ROC setting, one aims to maximize the true positive rate $\text{TPR} = \Pr(\hat{y} = 1|y = 1)$ while minimizing the false positive rate $\text{FPR} = \Pr(\hat{y} = 1|y = 0)$. The TPR can be calculated in PU data, by using the labeled positive set. While the FPR cannot be calculated from PU data, for a given TPR, minimizing the FPR within a hypothesis space $\mathcal{H}$ is equivalent to minimizing the probability of predicting the positive class $Pr(\hat{y} = 1)$:

$$\begin{aligned}\min_{\hat{y}:\mathcal{H},\text{TPR}} \Pr(\hat{y} = 1) &= \min_{\hat{y}:\mathcal{H},\text{TPR}} \alpha \Pr(\hat{y} = 1|y = 1) + (1 - \alpha) \Pr(\hat{y} = 1|y = 0) \\ &= \min_{\hat{y}:\mathcal{H},\text{TPR}} \alpha\text{TPR} + (1 - \alpha) \Pr(\hat{y} = 1|y = 0) \\ &= \alpha\text{TPR} + (1 - \alpha) \min_{\hat{y}:\mathcal{H},\text{TPR}} \Pr(\hat{y} = 1|y = 0).\end{aligned}$$

If classifier f exists that minimizes the FPR to zero, then the class prior can be calculated as $\alpha = \Pr(f = 1)/TPR = \Pr(f = 1)/\Pr(f = 1|s = 1)$. In fact, for any classifier f, this is an upper bound:

$$\alpha \geq \frac{\Pr(f=1)}{\Pr(f=1|s=1)}.$$

As a result, maximizing $\Pr(f=1)/\Pr(f=1|s=1)$ over the space of all classifiers gives the class prior (Blanchard et al. 2010). This result is valid under the irreducibility assumption. However, without extra assumptions, infinite examples are required for convergence. The stricter positive subdomain assumption allows for practical algorithms. Scott (2015) implements this idea by building a conditional probability classifier. The same idea is approached from a different angle by Jain et al. (2016; 2016). They use k-kernel density estimation to approximate the positive and total distributions, given different values for the class prior α, in a second step, they select α as the largest value (i.e., minimal $\Pr(\hat{y}=1)$ and thus minimal FPR) that results in the optimal log likelihood for both densities (i.e., maximal TPR).

6.5 Kernel embeddings

All previous methods, except TI*c*E, aim to model the entire domain with either discriminative or generative models. However, this might be overkill for estimating one constant, especially since the label frequency is equal for every example. Based on this insight, a class prior estimation method using kernel embeddings is proposed that aims to separate part of the positive distribution from the total distribution, under the positive function assumption. This means that they look for functions that map all negative examples to zero. Given a class prior, the minimal proportion from the negative distribution that is selected by any function is estimated. The class prior is the largest value for which that proportion is below a given threshold (Ramaswamy et al. 2016).

6.6 Other sources for the class prior

Estimating the class prior from PU data is hard. Therefore, it can be useful to obtain it in another way. For some domains, the class prior can be known from domain knowledge or previous studies. If there is access to a smaller dataset for the same domain that does have both possible and unlabeled labels, these can be used to estimate the class prior from. Or finally, one can just not estimate it but treat is as a hyperparameter and use a validation set and tune for it using a PU evaluation metric from Sect. 4.

6.7 Comparison of prior estimation methods

It is natural to wonder about the relative strengths and weaknesses of the various approaches for estimating the class prior. Whether a particular approach is suitable for a problem will depend on the assumptions underpinning the approach and how well they match the problem at hand. The non-traditional classifier (Elkan and Noto 2008; Northcutt et al. 2017) and some partial matching (Du Plessis and Sugiyama 2014) approaches make the assumption that the positive and negative example are separable. It is unlikely that this assumption will hold in practice. It is possible to relax this restriction for the partial matching (du Plessis et al. 2015b) approach such that only a positive subdomain is assumed. Moreover, this work is supported by theoretical analysis in terms of uniform deviation bounds and error estimation bounds. The decision tree approach TI*c*E and Jain et al.'s ROC

approach (Jain et al. 2016) also make this same assumption, but do not provide guarantees in terms of convergence to the true estimate. The kernel embedding approaches KM1 and KM2 (Ramaswamy et al. 2016) make the even less restrictive positive function assumption. Moreover, the work provides a proof that their algorithm for estimating the prior converges to the true prior under certain assumptions.

Empirically, the comparisons among these approaches tend to focus on idealized conditions on artificially constructed PU data. Hence, which approach is best in practice is still an important open issue. That being said, there are still some insight to be gleaned based on several recent studies. Bekker and Davis (2018a) compared canonical examples of each of aforementioned classes of approaches for estimating the class prior (apart from the techniques in Sect. 6.5). Using a small benchmark (11 datasets) under a number of different SCAR settings, they found that the kernel embedding approach KM2 (Ramaswamy et al. 2016) and TI*c*E (Bekker and Davis 2018a) produced the most accurate estimates on SCAR PU data. TI*c*E conferred the advantage of being significantly faster at estimating the class prior. In fact, it was only feasible to run KM2 on small subsets of the data. Of course, KM2 offers the advantage of having stronger theoretical underpinnings. Moreover, recently it was shown that KM2 results in more accurate classifier performance than TI*c*E on SAR PU data (Bekker et al. 2019).

7 Sources of PU data and applications

There are many classification situations where PU data naturally occurs and various machine learning tasks can be phrased as PU learning problems. The following subsection lists some of these situations and tasks. Next, applications that were explicitly addressed as PU learning problems are discussed.

7.1 Sources of PU data

PU data naturally arises in the following settings.

An *automatic diagnosis* system aims to predict if a patient has a disease. The data for such a system would consist of patients that were diagnosed with the disease and patients that were not. However, not being diagnosed is not equal to not having it. Many diseases, like diabetes, often go undiagnosed (Claesen et al. 2015c). Diagnoses patients are thus positive examples, while undiagnosed are unlabeled.

Sometimes, *positive examples are easier to obtain*. Recommendation systems, for example, can use previous purchases or likes as examples for items of interest. Similarly, some spam mails will be tagged as such. Purchased or tagged items are thus positive examples, while the others are unlabeled.

Indirect labels can be used to get some labeled examples. For example, to classify active students based on university records, the students that are registered in university sport classes are active. Other students are unlabeled.

The *case-control* scenario comes from the setting where two datasets are used and one is known to only have positive examples. For example, to predict one's socioeconomic status from her health record, positive examples could be gathered from health centers in upper-class neighborhoods and unlabeled examples from a random selection of health centers.

Negative-class dataset shift occurs when the distribution of the negative examples changes while the positive distribution remains the same. This happens, for example, in

adversarial scenarios. In this case it might be easier to obtain a new representative sample from the entire distribution than to label characteristic examples from the new negative distribution (Du Plessis et al. 2015a).

In surveys, *under-reporting* occurs when participants are likely to give false negative responses (Sechidis et al. 2017). This occurs for issues that have social stigma, such as maternal smoking. Research has shown that smoking may be underestimated by up to 47% (Gorber et al. 2009). In this setting, a negative response is really an unlabeled example.

The goal of *one-class classification* is to recognize examples from the class of interest, i.e., the positive class, from the entire population. When an unlabeled dataset is available that represents the entire population, then this can be seen as learning from positive and unlabeled data (Khan and Madden 2014). In this case, the negative class often has a large variety, for which it is difficult to label a representative sample (Li et al. 2011).

Inlier-based outlier detection has access to a representative sample of inliers, in addition to the standard unsupervised data. With this information, more powerful outlier detection is possible (Hido et al. 2008; Smola et al. 2009). This task can be phrased as PU learning, with the inliers as the positive class (Blanchard et al. 2010).

Automatic *knowledge base completion* is inherently a positive and unlabeled problem. Automatically constructed knowledge bases are necessarily incomplete and only contain true facts (Galárraga et al. 2015; Neelakantan et al. 2015). The unlabeled examples are the facts that are considered to be added to the knowledge base.

Identification problems aim to identify examples in an unlabeled dataset that are similar to the provided examples. For example, disease gene identification aims to identify new disease-genes (Mordelet and Vert 2011).

7.2 Applications

PU learning has been applied to a variety of problems.

Disease gene identification aims to identify which genes from the human genome are causative for diseases. Here, all the known disease genes are positive examples, while all other candidates, that can be generated by traditional linkage analysis, genes are unlabeled. To check all of the candidates individually would be very costly. With PU learning, a promising subset can be discovered. Several PU methods were developed to this end: ProDiGe is a method based on bagging SVMs (Mordelet and Vert 2011, 2014), PUDI is also a weighted SVM method, but they have different weights for four identified groups of unlabeled examples: reliable negative, likely positive, likely negative and weakly negative (Yang et al. 2012), EPU uses multiple biological data sources and trains an ensemble model on those (Yang et al. 2014).

Protein complexes are a set of interacting proteins for specific biological activities. Such complexes can be predicted as subgraphs from protein-protein interaction networks. Known complexes are positive examples and all other possibilities are unlabeled. This problem has been addressed using a non-traditional classifier approach (Elkan and Noto 2008; Zhao et al. 2016).

A *gene regulatory network* is a set of interacting genes that control cell functions. Using the non-traditional classifier method with SVMs, the relationships between activation profiles of gene pairs can be identified (Elkan and Noto 2008; Cerulo et al. 2010). Bagging SVMs have been employed to identify which genes are under control of which transcription factors (Mordelet and Vert 2014, 2013).

In the field of *drug discovery*, the tasks of *drug repositioning*, which looks for interactions between drugs and diseases, and *drug-drug-interactions* are very important. To find these interactions, a pairwise scoring function can be trained so that known interactions score higher than pairs which are not known to interact (Liu et al. 2017). The rationale behind this method is similar to RSVM (Sellamanickam et al. 2011).

Ecological modeling of the habitat of species aims to model where certain animals appear. An observed animal at a certain location provides positive examples. However not observing an animal does not mean that it never comes there. An EM algorithm on top of logistic regression that finds the optimal likelihood model, given the class prior, was proposed to address this application (Ward et al. 2009).

The goal of *targeted marketing* is to only promote products to potential buyers. The difficulty is to identify these customers. A biased SVM approach has been used to identify heat pump owners based on smart meter data, prior sales and weather data (Liu et al. 2003; Fei et al. 2013). For online retail, purchase data is often used as positive examples. However, for durable goods, like televisions, only a small fraction of potential customers will purchase it, not because they are not interested, but because already have one or are waiting for the right time, etc. A custom algorithm was developed for this application (Yi et al. 2017).

Remote sensing data, like satellite pictures, can be used to classify certain areas. While examples can be given for the class of interest, it can be hard to identify negative examples, because those are too diverse to be labeled. A non-traditional classifier can be used in such a context (Elkan and Noto 2008; Li et al. 2011).

Local descriptors play an important role in *localization* of, for example, mobile robots from laser scanner data. However, in some natural environment, many of the local descriptors might be unreliable and are better filtered out than used. To this end, the non-traditional random forest can be used, where the unlabeled examples are subsampled in a similar way as for bagging SVMs (Elkan and Noto 2008; Mordelet and Vert 2014; Breiman 2001; Latulippe et al. 2013).

Recommender systems can suffer from deceptive reviews, which are dishonest positive or negative reviews. These reviews should therefore be filtered out. Some positive examples of such reviews can be provided, but all other reviews to be checked are unlabeled (Ren et al. 2014).

Focused web crawlers search for relevant web pages given a query. Such a web crawler chooses to follow a link or not, based on the link's context. It is much easier to provide positive examples of such contexts than to provide a good sample of negative examples. Therefore the WVC and PSOC methods have been used to address this problem (Peng et al. 2007).

In *time series anomaly detection*, the goal is to identify portions of the data characterized by presence of unexpected or abnormal behavior. In the case of water usage data (Vercruyssene et al. 2018), recognizing certain patterns can play an important role in an anomaly detector. Because it is too time consuming to annotate all pattern occurrences in the data, an expert will typically annotate a few segments containing the pattern. The task of identifying the remaining patterns (Vercruyssen et al. 2020) can be viewed as a PU problem with the annotated segments serving as positive examples and unannotated segments as unlabeled examples, as these may or may not contain the pattern. The inductive bagging SVM (Mordelet and Vert 2014) has been shown to work well for this task.

8 Related fields

This section briefly discusses the fields that are closely related to PU learning.

8.1 Semi-supervised learning

The goal of semi-supervised learning is to learn from labeled and unlabeled data (Chapelle et al. 2009). In contrast to PU learning, labeled examples of all classes are assumed to be present in the data. Also, semi-supervised learning can go beyond binary classification tasks. Although semi-supervised methods cannot be applied directly to PU learning, some approaches have been ported from one domain to the other (Denis et al. 2003; Pelckmans and Suykens 2009).

For semi-supervised learning methods that incorporate the class prior, it is usually assumed that the class prior can be readily estimated from the labeled data, i.e., that positive and negative examples are selected to be labeled with the same probability. However, recently a matching method has been proposed to estimate the class prior when this is not the case (du Plessis and Sugiyama 2012).

8.2 One-class classification

The goal of one-class classification is to learn a model that identifies examples from a certain class: the positive class, when only examples of that class are available (Khan and Madden 2014). It can be seen as training a binary classifier where the negative class consists of all other possible classes. This is in contrast to PU learning, where the domain of interest is defined by the unlabeled data. Also, the unlabeled data enables finding low-density areas which are likely to be classification boundaries under the separability assumption. Under the SCAR assumption, areas with relatively more unlabeled examples than positive ones indicate a negative region, which would not be clear with only positive examples.

8.3 Classification in the presence of label noise

Label noise occurs when some of the class labels in the data are erroneous, i.e., when some examples have a class label that does not correspond with its true class value. A common interpretation of PU learning is that it is the specific type of label noise, called *one-sided label noise,* where the positive examples can be incorrectly labeled as negative (Scott et al. 2013). All the biased learning methods are based on this interpretation.

Just like the SCAR assumption was proposed in analogy with the MCAR assumption from missing data, a taxonomy for mislabeling mechanisms was proposed in analogy with the missing data taxonomy (Frénay and Verleysen 2014):

NCAR *Noisy Completely At Random* Every class label has exactly the same probability to be erroneous, independent of the attribute values of the example or the true class value.

NAR *Noisy At Random* The probability for a class label to be erroneous depends completely on the true class value, this is also known as asymmetric label noise.

NNAR *Noisy Not At Random* The probability for a class label to be erroneous depends on the attribute values

The SCAR labeling mechanism corresponds to the NAR mislabeling mechanism, where the mislabeling probability for the positive and negative class are $1 - c$ and 0 respectively. The label noise literature refers to mislabeling probability $1 - c$ as the noise rate or flip rate ρ_{+1} (Scott et al. 2013; Natarajan et al. 2013).

Because SCAR PU Learning is a specific setting of learning with NAR noisy labels, the SCAR methods can often be generalized to NAR. For example, rebalancing methods, where the instances get class-dependent weights, and empirical-risk-minimization based methods both exists for learning with NAR noisy labels (Natarajan et al. 2013, 2017). Rank pruning was also proposed for the general NAR noisy labels setting (Northcutt et al. 2017).

8.4 Missing data

When working with missing data, the missingness mechanism that dictates which values are missing plays a crucial role, just like the labeling mechanism for PU learning. The missingness mechanisms are generally divided into three classes (Rubin 1976; Little and Rubin 2002):

MCAR *Missing Completely At Random* Every attribute has exactly the same probability to be missing, independent of the other attribute values of the example and the value of the missing attribute.

MAR *Missing At Random* The probability for an attribute to be missing depends completely on the observable attributes of the example.

MNAR *Missing Not At Random* The probability for an attribute to be missing depends on the value that is missing.

The SCAR and SAR assumptions were introduced in analogy with MCAR and MAR. However, it is important to note that within the missing data taxonomy, SCAR and SAR actually both belong to the MNAR class, because positive and negative class values have a different probabilities to be missing: c or $e(x)$ and 0 respectively. The class values are missing (completely) at random only if just the population of positive examples is considered. Moreno et al. (2012) proposed a new missingness class: *Missing Completely At Random-Class Dependent (MAR-C)*, where per class, the data is MCAR, as is the case for SCAR.

8.5 Multiple-instance learning

The goal of multiple-instance learning is to train a binary classifier. Instead of positive and negative examples, the learner is provided with bags that are labeled positive if at least one of the examples in the bag is positive and negative otherwise. This setting can be phrased as PU learning, or actually NU learning, as the classes are switched. All the examples in a negative bag are known to be negative and can therefore get a negative label, while examples in a positive bag can be both positive and negative and therefore are considered

unlabeled. Following this insight, classifiers from either domains can be used to solve the task of the other domain (Li et al. 2013).

9 Conclusions and perspectives

PU learning is a very active area of research within the machine learning community. We will end by tying the survey back to the central PU learning research questions and discussing key future directions.

9.1 Questions revisited

At the end of the introduction, we posed seven research questions frequently addressed in PU learning research. To conclude, we will revisit these questions and try to synthesize answers to each one.

How can we formalize the problem of learning from PU data? The PU learning literature always assumes one of two learning scenarios: single-training-set or case-control, which are discussed in Section 2. The former assumes one dataset that is an i.i.d. sample of the true distribution. A subset of the positive examples of the dataset are labeled while the remaining examples are unlabeled. The latter scenario assumes two independently drawn datasets: an i.i.d. sample of the true distribution (unlabeled) and a sample of the positive part of the true distribution (positive). The labeled examples are selected from the positive subset or the positive distribution according to the labeling mechanism.

What assumptions are typically made about PU data in order to facilitate the design of learning algorithms? As discussed in Sect. 3, assumptions are needed either about the data distribution, or the labeling mechanism, or both. The most common assumptions about the data distribution are separable classes and smoothness, which form the basis for the two-step learning techniques. The most common labeling mechanism assumption is selected completely at random (SCAR) assumption, where postures that the set of labeled examples is a uniformly random subset of the positive examples. It greatly simplifies learning and it serves as the basis of all class-prior based methods. Recently, the more realistic SAR assumption has been proposed which assumes that the labeling mechanism depends on the attributes.

Can we estimate the class prior from PU data and why is this useful? By making assumptions about the data and/or labeling mechanism it is possible to estimate the label frequency and hence class prior in certain conditions (Sect. 3.3). Multiple different techniques have been proposed for this task (Sect. 6). The power and usefulness of this piece of information is that facilitates the design of algorithms for learning from PU data (Sect. 5.3). This is effectively done by estimating the expected number of positive and negative examples of the data, which can be accomplished by either weighting the data and then applying standard algorithms or directly modifying algorithms to work with fractional counts.

How can we learn a model from PU data? Section 5 shows that most PU learning methods belong to one of three categories: two-step techniques, biased learning and class prior incorporation methods. Two-step techniques begin by identifying reliable negative (and sometimes positive) examples and then using the labeled and reliable examples to train a classifier. The biased methods treat the unlabeled examples as belonging to the negative

class, but attribute a larger loss to false positives than false negatives. Class prior incorporation methods use the class prior to weight the unlabeled data or modify machine learning algorithms to reason about the expected number of positive and negative examples in the unlabeled data.

How can we evaluate models in a PU setting? This is an area that has perhaps received less attention in the literature. This can be approached in two general ways, both of which exploit the SCAR assumption. One is to use the (estimated) class prior and construction bounds for traditional evaluation metrics such as accuracy. The other is to design metrics that can be computed based on the observed information (e.g., could be computed using only positive examples) which are proxies for standard metrics. This was discussed in Sect. 4.

When and why does PU data arise in real-world applications? As outlined in Sect. 7, PU data arises in many different fields. At a high-level, it occurs in the following types of situations:

1. When only "positive" information is recorded such as in an electronic medical record or a knowledge base that stores facts, where the absence of information does not imply something is not true;
2. People have a reason to be deceptive and not report such as lying about smoking when pregnant in a survey or an athlete hiding an injury in order to keep playing;
3. Where it is much easier to identify one class than another, such as certain bioinformatics problems or remote sensing.

How does PU learning relate to other areas of machine learning Section 8 shows that PU learning is related to numerous areas of machine learning. Most obviously, it is a special case of standard semi-supervised learning. The key differences are that typically semi-supervised approaches have access to at least some examples of all classes, and that semi-supervised approaches go beyond binary classification tasks. Similarly, it can also be viewed through the prism of learning with label noise. Again, PU learning is a specialization in that corresponds to one type of noise: that where positive examples are possibly incorrectly labeled as negative. Some of the nomenclature about labeling mechanisms has been inspired by the long standing field of working with missing data. Finally, it also tied to one-class classification, learning with missing data and multiple-instance learning.

9.2 Future directions

Given that PU data naturally arises in many real-world datasets, it should continue to be an active area of machine learning research. The key open questions will revolve around making sure the assumptions and settings considered within PU learning align with real-world PU tasks. Therefore, there are several key directions that PU could take, which we now expand upon.

More realistic labeling mechanisms and corresponding learning methods One important area of research is to consider more realistic assumptions about the labeling mechanism. Until this year, the vast majority of work had focused on the SCAR assumption, given that it facilitates analysis. However, this assumption clearly often does not hold in practice. On the other side of the spectrum, there is the SAR assumption, which is so general that it essentially always holds. However, it is so general that effective learning in this

setting requires making additional assumptions. The probabilistic gap assumption finds some middle-ground. However, it does not always apply. For example, a professional sports player (e.g., a football or soccer player) in a contract year may be less likely to report a minor injury, but this has no relationship with the probability of a player getting injured. Therefore, researchers should continue to consider how to formalize different labeling assumptions that more closely resemble how PU data naturally arises within real-world applications. Additionally, learning methods should be developed that leverage these labeling assumptions.

An empirical comparison of PU learning approaches As this survey shows, a wide variety of PU learning approaches have been proposed. While many of the approaches have a strong theoretical basis, presuming certain assumptions hold, we still lack a complete empirical understanding of how the various approaches perform. In the literature, papers typically compare a hand full of approaches on a small number of datasets (i.e., often less than ten). Moreover, the considered datasets vary by paper. An extensive evaluation could help provide us with more insight into which methods are preferred and which assumptions are reasonable for obtaining good performance in practice.

Evaluating classifier performance on PU data The standard approach to evaluating a PU classifier's generalization ability is to assume a fully labeled test set. While this is convenient, it does not conform to the motivation of learning from PU data. There has been some work on evaluating classifier performance using PU data, which is a more challenging setting. However, much of this work is theoretical, and there has been little (if any) direct quantitative comparison among the various approaches (e.g., Claesen et al. 2015a; Jain et al. 2017; Sechidis et al. 2014). An important future direction is understanding how these metrics perform in practice. Furthermore, often these approaches rely on the SCAR assumption (e.g., Claesen et al. 2015a; Jain et al. 2017) and it will be important to design metrics that work for other labeling mechanisms.

Real-world PU benchmarks The current evaluation paradigm largely consists of using existing, fully labeled datasets and converting them into a PU setting. This has advantages and disadvantages. The positive aspect is it provides a controlled manner in which to assess performance. This setup typically ensures that the assumptions made in the paper are respected. The disadvantage is that we then lack an understanding about what will happen "in the wild" when the assumptions are violated. One partial remedy would be to encourage authors to simulate these violations. Ideally, several real-world PU benchmarks could be created and released, which would greatly benefit the community. We do note that in the fully PU setting, evaluation would be very tricky. One promising domain for this is knowledge base completion. While this is often not view through the lens of PU learning, the task certainly could be categorized in this way.

PU learning in relational domains The vast majority of PU learning work has focused on the propositional setting. There has been a renewed interest recently in learning from relational data. This dovetails with the previous suggestion in that knowledge base completion is inherently a relational problem. Therefore, it may be fruitful to further explore how to enable PU learning in relational domains both from a theoretical and algorithmic perspective.

Acknowledgements JB is supported by IWT (SB/141744). JD is partially supported by FWO-Vlaanderen (G0D8819N), KU Leuven Research Fund (C14/17/070), and the Flemish Government under the "Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen" programme.

References

Basile, T. M., Di Mauro, N., Esposito, F., Ferilli, S., & Vergari, A. (2018). Density estimators for positive-unlabeled learning. In *New frontiers in mining complex patterns: 6th international workshop, NFMCP 2017, held in conjunction with ECML-PKDD 2017*, Skopje, Macedonia, September 18–22, 2017, Revised Selected Papers (Vol. 10785, pp. 49–64). Berlin: Springer.

Bekker, J., & Davis, J. (2018a). Estimating the class prior in positive and unlabeled data through decision tree induction. In *Proceedings of the 32th AAAI conference on artificial intelligence* (pp. 2712–2719).

Bekker, J., & Davis, J. (2018b). Positive and unlabeled relational classification through label frequency estimation. In N. Lachiche & C. Vrain (Eds.), *Inductive logic programming* (pp. 16–30). Cham: Springer.

Bekker, J., Robberechts, P., & Davis, J. (2019). Beyond the selected completely at random assumption for learning from positive and unlabeled data. In *ECML PKDD: Joint European conference on machine learning and knowledge discovery in databases*. Cham: Springer.

Blanchard, G., Lee, G., & Scott, C. (2010). Semi-supervised novelty detection. *Journal of Machine Learning Research*, *11*, 2973–3009.

Blockeel, H. (2017). Pu-learning disjunctive concepts in ilp. In *ILP 2017 late breaking papers*.

Blum, A., & Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory* (pp. 92–100). ACM.

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32. https://doi.org/10.1023/A:1010933404324.

Calvo, B., Larrañaga, P., & Lozano, J. A. (2007). Learning bayesian classifiers from positive and unlabeled examples. *Pattern Recognition Letters*, *28*(16), 2375–2384. https://doi.org/10.1016/j.patrec.2007.08.003.

Cerulo, L., Elkan, C., & Ceccarelli, M. (2010). Learning gene regulatory networks from only positive and unlabeled data. *BMC Bioinformatics*, *11*(1), 228.

Chang, S., Zhang, Y., Tang, J., Yin, D., Chang, Y., Hasegawa-Johnson, M. A., & Huang, T. S. (2016). Positive-unlabeled learning in streaming networks. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 755–764). ACM.

Chapelle, O., Scholkopf, B., & Zien, A. (2009). Semi-supervised learning. *IEEE Transactions on Neural Networks*, *20*(3), 542–542.

Chaudhari, S., & Shevade, S. (2012). Learning from positive and unlabelled examples using maximum margin clustering. In *Proceedings of the 19th international conference on neural information processing* (Vol. 3, pp. 465–473). Berlin, Heidelberg: Springer.

Chiaroni, F., Rahal, M. C., Hueber, N., & Dufaux, F. (2018). Learning with a generative adversarial network from a positive unlabeled dataset for image classification. In *IEEE international conference on image processing*.

Claesen, M., Davis, J., De Smet, F., & De Moor, B. (2015a). Assessing binary classifiers using only positive and unlabeled data. arXiv preprint arXiv:1504.06837.

Claesen, M., De Smet, F., Gillard, P., Mathieu, C., & De Moor, B. (2015b). Building classifiers to predict the start of glucose-lowering pharmacotherapy using Belgian health expenditure data. arXiv preprint arXiv:1504.07389.

Claesen, M., Smet, F. D., Gillard, P., Mathieu, C., & Moor, B. D. (2015c). Building classifiers to predict the start of glucose-lowering pharmacotherapy using Belgian health expenditure data. CoRR arXiv:abs/1504.07389.

Claesen, M., Smet, F. D., Suykens, J. A. K., & Moor, B. D. (2015d). A robust ensemble approach to learn from positive and unlabeled data using SVM base models. *Neurocomputing*, *160*, 73–84.

Denis, F., Gilleron, R., & Letouzey, F. (2005). Learning from positive and unlabeled examples. *Theoretical Computer Science*, *348*(1), 70–83.

Denis, F., Laurent, A., Gilleron, R., & Tommasi, M. (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* (pp. 80–87).

du Plessis, M. C., Niu, G., & Sugiyama, M. (2014). Analysis of learning from positive and unlabeled data. In *Advances in neural information processing systems* (pp. 703–711).

Du Plessis, M., Niu, G., & Sugiyama, M. (2015a). Convex formulation for learning from positive and unlabeled data. In *International conference on machine learning* (pp. 1386–1394).

du Plessis, M., Niu, G., & Sugiyama, M. (2015b). Class-prior estimation for learning from positive and unlabeled data. In *Proceedings of the 7th Asian conference on machine learning* (pp. 221–236).

du Plessis, M. C., & Sugiyama, M. (2012). Semi-supervised learning of class balance under class-prior change by distribution matching. *Neural Networks: The Official Journal of the International Neural Network Society*, *50*, 110–9.

Du Plessis, M. C., & Sugiyama, M. (2014). Class prior estimation from positive and unlabeled data. *IEICE Transactions on Information and Systems*, *97*(5), 1358–1362.

Elkan, C. (2001). The foundations of cost-sensitive learning. In *Proceedings of the seventeenth international joint conference on artificial intelligence* (Vol. 17, pp. 973–978). Lawrence Erlbaum Associates Ltd.

Elkan, C., & Noto, K. (2008). Learning classifiers from only positive and unlabeled data. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 213–220). ACM.

Fei, H., Kim, Y., Sahu, S., Naphade, M., Mamidipalli, S. K., & Hutchinson, J. (2013). Heat pump detection from coarse grained smart meter data with positive and unlabeled learning. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 1330–1338). ACM.

Frénay, B., & Verleysen, M. (2014). Classification in the presence of label noise: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, *25*(5), 845–869.

Fung, G. P. C., Yu, J. X., Lu, H., & Yu, P. S. (2006). Text classification without negative examples revisit. *IEEE Transactions on Knowledge and Data Engineering*, *18*, 6–20.

Galárraga, L., Teflioudi, C., Hose, K., & Suchanek, F. M. (2015). Fast rule mining in ontological knowledge bases with AMIE+. *The International Journal on Very Large Data Bases*, *24*(6), 707–730.

Gan, H., Zhang, Y., & Song, Q. (2017). Bayesian belief network for positive unlabeled learning with uncertainty. *Pattern Recognition Letters*, *90*(C), 28–35. https://doi.org/10.1016/j.patrec.2017.03.007.

Gorber, S. C., Schofield-Hurwitz, S., Hardt, J. S., Levasseur, G., & Tremblay, M. D. (2009). The accuracy of self-reported smoking: A systematic review of the relationship between self-reported and cotinine-assessed smoking status. *Nicotine & Tobacco Research : Official Journal of the Society for Research on Nicotine and Tobacco*, *11*(1), 12–24.

He, F., Liu, T., Webb, G. I., & Tao, D. (2018). Instance-dependent pu learning by Bayesian optimal relabeling. arXiv preprint arXiv:1808.02180.

He, J., Zhang, Y., Li, X., & Wang, Y. (2010). Naive bayes classifier for positive unlabeled learning with uncertainty. In *Proceedings of the 2010 SIAM international conference on data mining* (pp. 361–372). SIAM.

He, J., Zhang, Y., Li, X., & Wang, Y. (2011). Bayesian classifiers for positive unlabeled learning. In *Proceedings of the 12th international conference on Web-age information management, WAIM'11* (pp. 81–93). Berlin, Heidelberg: Springer. http://dl.acm.org/citation.cfm?id=2035562.2035574.

Hido, S., Tsuboi, Y., Kashima, H., Sugiyama, M., & Kanamori, T. (2008). Inlier-based outlier detection via direct density ratio estimation. In *2008 Eighth IEEE international conference on data mining* (pp. 223–232).

Hou, M., Chaib-draa, B., Li, C., & Zhao, Q. (2018). Generative adversarial positive-unlabelled learning. In *Proceedings of the twenty-seventh international joint conference on artificial intelligence, IJCAI-18* (pp. 2255–2261). https://doi.org/10.24963/ijcai.2018/312.

Hsieh, C. J., Natarajan, N., & Dhillon, I. (2015). PU learning for matrix completion. In *International conference on machine learning* (pp. 2445–2453).

Ienco, D., & Pensa, R. G. (2016). Positive and unlabeled learning in categorical data. *Neurocomputing*, *196*(C), 113–124. https://doi.org/10.1016/j.neucom.2016.01.089.

Ienco, D., Pensa, R. G., & Meo, R. (2012). From context to distance: Learning dissimilarity for categorical data clustering. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, *6*(1), 1–25.

Jain, S., White, M., & Iivojac, P. (2016). Estimating the class prior and posterior from noisy positives and unlabeled data. In *Advances in neural information processing systems* (pp. 2693–2701).

Jain, S., White, M., & Radivojac, P. (2017). Recovering true classifier performance in positive-unlabeled learning. In *Proceedings of the 31st AAAI conference on artificial intelligence* (pp. 2066–2073).

Jain, S., White, M., Trosset, M. W., & Radivojac, P. (2016). Nonparametric semi-supervised learning of class proportions. arXiv preprint arXiv:1601.01944.

Jiang, L., Zhang, H., & Cai, Z. (2009). A novel bayes model: Hidden naive bayes. *IEEE Transactions on Knowledge and Data Engineering*, *21*(10), 1361–1371.

Ke, T., Jing, L., Lv, H., Zhang, L., & Hu, Y. (2017). Global and local learning from positive and unlabeled examples. *Applied Intelligence*, *48*, 2373–2392.

Ke, T., Lv, H., Sun, M., & Zhang, L. (2018). A biased least squares support vector machine based on Mahalanobis distance for PU learning. *Physica A: Statistical Mechanics and its Applications*, *509*, 422–438. https://doi.org/10.1016/j.physa.2018.05.128.

Ke, T., Yang, B., Zhen, L., Tan, J., Li, Y., & Jing, L. (2012). Building high-performance classifiers using positive and unlabeled examples for text classification. In *International symposium on neural networks* (pp. 187–195). Berlin: Springer.

Khan, S., & Madden, M. (2014). One-class classification: Taxonomy of study and review of techniques. *The Knowledge Engineering Review*.

Khot, T., Natarajan, S., & Shavlik, J. W. (2014). Relational one-class classification: A non-parametric approach. In *Proceedings of the 28th AAAI conference on artificial intelligence* (pp. 2453–2460).

Kiryo, R., Niu, G., du Plessis, M. C., & Sugiyama, M. (2017). Positive-unlabeled learning with non-negative risk estimator. In *Advances in neural information processing systems* (pp. 1675–1685).

Kull, M., de Menezes e Silva Filho, T., & Flach, P. A. (2017). Beta calibration: A well-founded and easily implemented improvement on logistic calibration for binary classifiers. In *Proceedings of the twentieth international conference on artificial intelligence and statistics* (pp. 623–631).

Latulippe, M., Drouin, A., Giguere, P., & Laviolette, F. (2013). Accelerated robust point cloud registration in natural environments through positive and unlabeled learning. In *Proceedings of the 23th international joint conference on artifical intelligence* (pp. 2480–2487).

Lee, W. S., & 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).

Li, W., Guo, Q., & Elkan, C. (2011). A positive and unlabeled learning algorithm for one-class classification of remote-sensing data. *IEEE Transactions on Geoscience and Remote Sensing*, *49*, 717–725.

Li, X., & Liu, B. (2003). Learning to classify texts using positive and unlabeled data. *Proceedings of the eighteenth International Joint Conference on Artifical Intelligence*, *3*, 587–592.

Li, X., Liu, B., & Ng, S. K. (2007). Learning to identify unexpected instances in the test set. In *Proceedings of the 20th international joint conference on artifical intelligence* (Vol. 7, pp. 2802–2807).

Li, X. L., & Liu, B. (2005). Learning from positive and unlabeled examples with different data distributions. In *European conference on machine learning* (pp. 218–229). Berlin: Springer.

Li, X. L., Liu, B., & Ng, S. K. (2010). Negative training data can be harmful to text classification. In *Proceedings of the 2010 conference on empirical methods in natural language processing* (pp. 218–228). Association for Computational Linguistics.

Li, X. L., Yu, P. S., Liu, B., & Ng, S. K. (2009). Positive unlabeled learning for data stream classification. In *Proceedings of the 2009 SIAM international conference on data mining* (pp. 259–270). SIAM.

Li, Y., Tax, D. M., Duin, R. P., & Loog, M. (2013). The link between multiple-instance learning and learning from only positive and unlabelled examples. In *International workshop on multiple classifier systems* (pp. 157–166). Berlin: Springer.

Liang, C., Zhang, Y., Shi, P., & Hu, Z. (2012). Learning very fast decision tree from uncertain data streams with positive and unlabeled samples. *Information Sciences*, *213*, 50–67.

Little, R. J., & Rubin, D. B. (2002). *Statistical analysis with missing data*. Hoboken: Wiley.

Liu, B., Dai, Y., Li, X., Lee, W. S., & Yu, P. S. (2003). Building text classifiers using positive and unlabeled examples. In *Proceedings of the third IEEE international conference on data mining* (pp. 179–186). IEEE.

Liu, B., Lee, W. S., Yu, P. S., & Li, X. (2002). Partially supervised classification of text documents. In *Proceedings of the nineteenth international conference on machine learning* (Vol. 2, pp. 387–394). Citeseer.

Liu, L., & Peng, T. (2014). Clustering-based method for positive and unlabeled text categorization enhanced by improved TFIDF. *Journal of Information Science and Engineering*, *30*, 1463–1481.

Liu, T., & Tao, D. (2016). Classification with noisy labels by importance reweighting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *38*, 447–461.

Liu, Y., Qiu, S., Zhang, P., Gong, P., Wang, F., Xue, G., & Ye, J. (2017). Computational drug discovery with dyadic positive-unlabeled learning. In *Proceedings of the 2017 SIAM international conference on data mining* (pp. 45–53). SIAM.

Liu, Z., Shi, W., Li, D., & Qin, Q. (2005). Partially supervised classification-based on weighted unlabeled samples support vector machine. In *Proceedings of the international conference on advanced data mining and applications* (pp. 118–129). Berlin: Springer.

Lu, F., & Bai, Q. (2010). Semi-supervised text categorization with only a few positive and unlabeled documents. In *2010 3rd International conference on biomedical engineering and informatics* (Vol. 7, pp. 3075–3079).

Mahalanobis, P. (1936). *On the generalised distance in statistics*. National Institute of Science of India.

Mordelet, F., & Vert, J. P. (2011). Prodige: Prioritization of disease genes with multitask machine learning from positive and unlabeled examples. *BMC Bioinformatics*, *12*, 389.

Mordelet, F., & Vert, J. P. (2013). Supervised inference of gene regulatory networks from positive and unlabeled examples. *Methods in Molecular Biology*, *939*, 47–58.

Mordelet, F., & Vert, J. P. (2014). A bagging svm to learn from positive and unlabeled examples. *Pattern Recognition Letters*, *37*, 201–209.

Muggleton, S. (1996). Learning from positive data. In *Selected papers from the 6th international workshop on inductive logic programming* (pp. 358–376).
Natarajan, N., Dhillon, I. S., Ravikumar, P., & Tewari, A. (2013). Learning with noisy labels. In *NIPS*.
Natarajan, N., Dhillon, I. S., Ravikumar, P., & Tewari, A. (2017). Cost-sensitive learning with noisy labels. *Journal of Machine Learning Research*, *18*, 155:1–155:33.
Natarajan, N., Rao, N., & Dhillon, I. (2015). PU matrix completion with graph information. In *2015 IEEE 6th international workshop on computational advances in multi-sensor adaptive processing (CAMSAP)* (pp. 37–40). IEEE.
Neelakantan, A., Roth, B., & McCallum, A. (2015). Compositional vector space models for knowledge base completion. In *Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing* (Vol. 1: Long Papers, pp. 156–166). Association for Computational Linguistics. https://doi.org/10.3115/v1/P15-1016. http://www.aclweb.org/anthology/P15-1016.
Nguyen, M. N., Li, X. L., Ng, S. K. (2011). Positive unlabeled learning for time series classification. In *Proceedings of the seventeenth international joint conference on artificial intelligence* (pp. 1421–1426).
Northcutt, C. G., Wu, T., & Chuang, I. L. (2017). Learning with confident examples: Rank pruning for robust classification with noisy labels. In *Proceedings of the thirty-third conference on uncertainty in artificial intelligence*, UAI'17. AUAI Press. http://auai.org/uai2017/proceedings/papers/35.pdf.
Pelckmans, K., & Suykens, J. A. (2009). Transductively learning from positive examples only. In *Proceedings of the European symposium on artificial neural networks* (pp. 23–28).
Peng, T., Zuo, W., & He, F. (2007). Svm based adaptive learning method for text classification from positive and unlabeled documents. *Knowledge and Information Systems*, *16*, 281–301.
Platt, J., et al. (1999). Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in Large Margin Classifiers*, *10*(3), 61–74.
Qin, X., Zhang, Y., Li, C., & Li, X. (2012). Learning from data streams with only positive and unlabeled data. *Journal of Intelligent Information Systems*, *40*, 405–430.
Ramaswamy, H., Scott, C., & Tewari, A. (2016). Mixture proportion estimation via kernel embedding of distributions. In *International conference on machine learning* (pp. 2052–2060).
Ren, Y., Ji, D., & Zhang, H. (2014). Positive unlabeled learning for deceptive reviews detection. In *Proceedings of the conference on empirical methods in Natural Language processing* (pp. 488–498).
Rubin, D. B. (1976). Inference and missing data. *Biometrika*, *63*(3), 581–592.
Scott, C. (2015). A rate of convergence for mixture proportion estimation, with application to learning from noisy labels. In *Proceedings of the 18th international conference on artificial intelligence and statistics* (pp. 838–846).
Scott, C., & Blanchard, G. (2009). Novelty detection: Unlabeled data definitely help. In *The 12th international conference on artificial intelligence and statistics* (pp. 464–471).
Scott, C., Blanchard, G., Handy, G., Pozzi, S., & Flaska, M. (2013). Classification with asymmetric label noise: Consistency and maximal denoising. In *Conference on learning theory*.
Sechidis, K., & Brown, G. (2015). Markov blanket discovery in positive-unlabelled and semi-supervised data. In *ECML PKDD: Joint European conference on machine learning and knowledge discovery in databases* (pp. 351–366). Berlin: Springer.
Sechidis, K., & Brown, G. (2017). Simple strategies for semi-supervised feature selection. *Machine Learning*, *107*, 357–395.
Sechidis, K., Calvo, B., & Brown, G. (2014). Statistical hypothesis testing in positive unlabelled data. In *ECML PKDD: Joint European conference on machine learning and knowledge discovery in databases*, (pp. 66–81). Berlin: Springer.
Sechidis, K., Sperrin, M., Petherick, E. S., Luján, M., & Brown, G. (2017). Dealing with under-reported variables: An information theoretic solution. *International Journal of Approximate Reasoning*, *85*, 159–177.
Sellamanickam, S., Garg, P., & Keerthi, S. S. (2011). A pairwise ranking based approach to learning with positive and unlabeled examples. In *Proceedings of the 2011 ACM on conference on information and knowledge management*.
Shao, Y. H., Chen, W. J., Liu, L. M., & Deng, N. Y. (2015). Laplacian unit-hyperplane learning from positive and unlabeled examples. *Information Sciences*, *314*, 152–168.
Smola, A. J., Song, L., & Teo, C. H. (2009). Relative novelty detection. In *The 12th international conference on artificial intelligence and statistics* (pp. 536–543).
Srinivasan, A. (2001). The Aleph manual.
Steinberg, D., & Scott Cardell, N. (1992). Estimating logistic regression models when the dependent variable has no variance. *Communications in Statistics-Theory and Methods*, *21*(2), 423–450.

Su, J., & Zhang, H. (2006). Full Bayesian network classifiers. In *Proceedings of the 23rd international conference on Machine learning* (pp. 897–904). ACM.

Suykens, J. A. K., & Vandewalle, J. (1999). Least squares support vector machine classifiers. *Neural Processing Letters*, *9*, 293–300.

Vercruyssen, V., Meert, W., & Davis, J. (2020). "now you see it, now you don't! "detecting suspicious pattern absences in continuous time series. In *Proceedings of the 2020 SIAM international conference on data mining*.

Vercruyssen, V., Wannes, M., Gust, V., Koen, M., Ruben, B., & Jesse, D. (2018). Semi-supervised anomaly detection with an application to water analytics. In *Proceedings/IEEE international conference on data mining*. IEEE.

Ward, G., Hastie, T., Barry, S., Elith, J., & Leathwick, J. R. (2009). Presence-only data and the em algorithm. *Biometrics*, *65*(2), 554–563.

Webb, G. I., Boughton, J. R., & Wang, Z. (2005). Not so naive Bayes: Aggregating one-dependence estimators. *Machine Learning*, *58*, 5–24.

Xu, Z., Qi, Z., & Zhang, J. (2014). Learning with positive and unlabeled examples using biased twin support vector machine. *Neural Computing and Applications*, *25*, 1303–1311.

Yang, P., Li, X., Chua, H. N., Kwoh, C. K., Ng, S. K. (2014). Ensemble positive unlabeled learning for disease gene identification. In *PloS ONE*.

Yang, P., Li, X., Mei, J. P., Kwoh, C. K., & Ng, S. K. (2012). Positive-unlabeled learning for disease gene identification. *Bioinformatics*, *28*, 2640–2647.

Yi, J., Hsieh, C. J., Varshney, K. R., Zhang, L., & Li, Y. (2017). Scalable demand-aware recommendation. In *Advances in neural information processing systems* (pp. 2412–2421).

Yu, H. (2005). Single-class classification with mapping convergence. *Machine Learning*, *61*(1–3), 49–69.

Yu, H., Han, J., & Chang, K. C. (2004). PEBL: Web page classification without negative examples. *IEEE Transactions on Knowledge and Data Engineering*, *16*(1), 70–81.

Yu, H., Han, J., & Chang, K. C. C. (2002). PEBL: positive example based learning for web page classification using svm. In *Proceedings of the eighth ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 239–248). ACM.

Yu, S., & Li, C. (2007). Pe-puc: A graph based pu-learning approach for text classification. In *International workshop on machine learning and data mining in pattern recognition* (pp. 574–584). Berlin: Springer.

Zadrozny, B., & Elkan, C. (2002). Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the eighth ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 694–699). ACM.

Zhang, B., & Zuo, W. (2009). Reliable negative extracting based on knn for learning from positive and unlabeled examples. *Journal of Computers*, *4*(1), 94–101.

Zhang, D., & Lee, W. S. (2005). A simple probabilistic approach to learning from positive and unlabeled examples. In *Proceedings of the fifth annual UK workshop on computational intelligence (UKCI)* (pp. 83–87).

Zhang, Y., Ju, X., & Tian, Y. (2014). Nonparallel hyperplane support vector machine for pu learning. In *2014 10th International conference on natural computation (ICNC)* (pp. 703–708).

Zhao, J., Liang, X., Wang, Y., Xu, Z., & Liu, Y. (2016). Protein complexes prediction via positive and unlabeled learning of the ppi networks. In *Proceedings of the 13th international conference on service systems and service management (ICSSSM)* (pp. 1–6). https://doi.org/10.1109/ICSSSM.2016.7538432.

Zhou, D., Bousquet, O., Lal, T. N., Weston, J., & Schölkopf, B. (2004). Learning with local and global consistency. *Advances in Neural Information Processing Systems*, *17*, 321–328.

Zhou, J. T., Pan, S. J., Mao, Q., & Tsang, I. W. (2012). Multi-view positive and unlabeled learning. In *Proceedings of the 4th Asian conference on machine learning*.

Zhou, K., Xue, G. R., Yang, Q., & Yu, Y. (2010). Learning with positive and unlabeled examples using topic-sensitive plsa. *IEEE Transactions on Knowledge and Data Engineering*, *22*, 46–58.

Zupanc, K., & Davis, J. (2018). Estimating rule quality for knowledge base completion with the relationship between coverage assumption. In *Proceedings of the Web conference* (pp. 1–9).