Group Members and Work Distribution
Liu Boyu (A0177847J) Liu Yu (A0177906R) Wang Zihong (A0230339L)
Wu Ruichi (A0177880N) Zhu Xinji (A0177866H)

Presentation and Report Contribution: Every team member

Code Contribution:
Liu Yu and Wang Zihong: VPU models, Liu Boyu: nnPU models and overall code collating
Wu Ruichi: nnPU models, Zhu Xinji: Raw NN models

1 Background of the Problem

1.1 PU Learning

In many real-life scenarios, people may commonly encounter situations where the data obtained consists of some positive samples and a large amount of unlabeled data. For instance, in biomedical research, positive instances of a protein phosphorylation site may be available, but negative examples are either restricted or ambiguous because it is far easier to confirm a property than to prove that it does not exist [6]. To enable the binary classifier to learn from restricted positive samples and a large number of combinations of positive or negative unlabeled samples, the positive unlabeled (PU) learning method was developed [3].

1.2 Existing Approaches of PU Learning

To address the unique challenges of PU learning, several models have been proposed in past years, such as Ranking Pruning (RP), Positive and Unlabeled learning with Label Disambiguation (PULD), and Generative Adversarial Networks (GAN). These models have demonstrated varying degrees of success in different applications but still have limitations.

Rank Pruning (RP) is an algorithm designed to estimate the noise rates of unlabeled examples [7]. Because it removes mislabeled instances before training, RP is sensitive to noise and relies on the quality of the initial ranking for optimal performance. PULD considers all unlabeled examples to be ambiguously labelled and then employs a margin-based label disambiguation technique to determine the unique ground-truth label of each unlabeled example [10]. It should be noted that PULD is only applicable to linear classifiers in non-trainable feature spaces. The third model GAN will produce fake positive and negative samples and use them for training the classifier [5]. However, GAN is difficult to train and unstable. It is only effective when the positive labelled data is very little [4].

All of the three algorithms involves the risk estimator [2], which calculates the risk of a classifier

Φ by

$$\mathcal{L}(\Phi) = \pi_P E_{\text{labeled data}}\left[l_+(\Phi(x)) - l_-(\Phi(x))\right] + E_{\text{unlabeled data}}\left[l_-(\Phi(x))\right] \tag{1}$$

where l_- and l_+ represent the misclassification loss on positive and negative data, and π_P represents the class prior.

Through empirical averaging, a risk estimator can arrive at a fair estimation of the predicted misclassification risk [2]. When class prior P is known, training the classifier by minimizing the estimated risk is feasible. However, it's worth noting that the class prior has some limitations. Unless sufficient negative data are present in the validation set, the class prior cannot be automatically picked as a trainable parameter and is difficult to modify as a hyper-parameter via cross-validation. As a result, it typically requires time-consuming class-prior estimation techniques like kernel machines.

1.3 A Variational Approaches for PU Learning

With the limitations addressed above, a variational approach for PU learning (VPU) was introduced. Using only distributions of labelled and unlabeled data, VPU compares a given classifier to the ideal Bayesian classifier in a class prior-free manner [1]. Without estimating the class prior, it can complete PU learning tasks with good classification accuracy.

2 Technical Approach

In our project, we reproduced the work of paper [1]. It is important to note that the theories presented in this section are derived from their paper, and we acknowledge their contribution.

2.1 Problem Setting

The problem we focus on is binary classification, where instances are described by real-number features x and class labels y. The instances are generated from a joint distribution P(x, y) and belong to one of two classes, -1 or +1. We are given a set of labelled instances P and an unlabeled set U. The task of PU learning is to construct a binary classifier that accurately predicts the class labels of unseen instances.

The approaches in this project are based on certain assumptions:

Assumption 1.

Labelled and unlabeled data are independently drawn as $P = \{x_i\}_{i=1}^{M} \overset{\text{iid}}{\sim} f_P$, $U = \{x_i\}_{i=M+1}^{M+N} \overset{\text{iid}}{\sim} f$, where M and N are the sizes of the labelled and unlabeled datasets, respectively. f_P is the distribution of the positive class and f is the marginal distribution of the instance feature.

Assumption 2.

There exists a set $A \subset R^d$ such that $\int_A f_P(x)dx > 0$ and $\Phi^*(x) = 1$ for all $x \in A$, where $\Phi^*(x)$ is the ideal Bayesian classifier. This implies that the set of x are almost surely positive.

2.2 Data

We evaluated our models on two datasets that were used in the paper: Page Blocks [8], and Fashion MNIST [9].

Page Blocks is a simple dataset consisting of 10 real-number features and a 5-class (denoted by 1-5) target variable. It is very imbalanced as 90% of the data belong to class 1. In this binary classification problem, we combine classes 2, 3, 4, and 5 together as one class.

FashionMNIST is a classic image dataset with 10 labels (denoted by 0-9). Data with labels 1, 4, and 7 are combined into one class, and the rest from the other class.

During implementation, we trained and tested the model twice on each dataset. Each time, we used part of the data from one class as positive, and all of the rest in the training dataset as unlabeled.

2.3 Variational PU (VPU) Learning

The paper introduces a new principle for PU learning that does not require the estimation of class priors. The principle is based on the Bayes rule and utilizes a parametric model Φ to approximate the positive data distribution f_P. Specifically, f_P can be approximated by

$$f_P(x) = \frac{P(y=+1|x)P(x)}{\int P(y=+1|x)P(x)dx} \approx \frac{\Phi(x)f(x)}{E_f[\Phi(x)]} \triangleq f_\Phi(x) \tag{2}$$

where $f_\Phi(x)$ is the density function of $\Phi(x)$.

Then the main theorem is based on the analysis of the approximation quality of Φ. This quality can be evaluated using the Kullback-Leibler (KL) divergence between f_P and f_Φ.

Theorem 3. For all $\Phi : R^d \to [0, 1]$ with $E_f[\Phi(x)] > 0$,

$$\mathrm{KL}(f_P||f_\Phi) = \mathcal{L}_{\mathrm{var}}(\Phi) - \mathcal{L}_{\mathrm{var}}(\Phi^*), \tag{3}$$

under Assumption 1, where $\mathcal{L}_{\mathrm{var}}(\Phi) \triangleq \log E_f[\Phi(x)] - E_{f_P}[\log \Phi(x)]$.

The paper explains that $L_{var}(\Phi)$ serves as a variational upper bound for $L_{var}(\Phi)$ due to the non-negativity of the KL divergence. Empirical averages over set P and U can be used to compute $L_{var}(\Phi^)$, and minimizing $L_{var}(\Phi)$ is equivalent to minimizing the KL divergence of f_P from f_Φ.

With a MixUp-based consistency regularization term, the parameters of Φ can be optimized through the minimization of $\mathcal{L}(\Phi) = \mathcal{L}_{\text{var}}(\Phi) + \lambda\mathcal{L}_{reg}(\Phi)$ (subject to constraints $\Phi(x) \in [0, 1]$ and $\max_x \Phi(x) = 1$).

The regularization term is defined as

$$\mathcal{L}_{reg}(\Phi) = E_{\tilde{\Phi},\tilde{x}}\left[\left(\log\tilde{\Phi} - \log\Phi(\tilde{x})\right)^2\right], \tag{4}$$

with $\gamma \overset{\text{iid}}{\sim} Beta(\alpha, \alpha)$, $\tilde{x} = \gamma \cdot x' + (1-\gamma)\cdot x''$, and $\tilde{\Phi} = \gamma \cdot 1 + (1-\gamma)\cdot\Phi(x'')$. where $\tilde{x}$ is a new sample that is created by combining two randomly selected samples, x' from the original dataset P and x'' from an external dataset U. $\tilde{\Phi}$ is the estimated probability of the positive class given $\tilde{x}$, which is obtained by linearly interpolating between the true label and the predicted label by the model Φ.

2.4 Baseline Models:

To validate the improvement made by VPU learning, we adopted two baseline models for each dataset, a Non-Negative Risk Estimator for PU Learning (nnPU) and a raw neural network (NN) model, and compared their performance.

2.4.1 Non-Negative Risk Estimator for PU Learning (nnPU)

nnPU, as proposed in [5], is a state-of-art PU learning model and is one of the baseline models used in the paper. It was developed based on an unbiased risk estimator.

In normal positive-negative learning problems, the loss is calculated as

$$\mathcal{L}_{pn} = \pi_p\mathcal{L}_p(+) + \pi_n\mathcal{L}_n(-) \tag{5}$$

where $\pi_p = p(Y = +1)$ is the *class-prior probability*, $\pi_n = p(Y = -1) = 1 - \pi_p$ and $\mathcal{L}_p(+)$, $\mathcal{L}_n(-)$ are the expected losses incurred by predicting on the positive data which are labelled as positive in the dataset and predicting on the negative data which are labelled as negative in the dataset.

However, in the PU learning problems, we have no negative data but unlabeled ones instead. In this case, we consider evaluating the loss for the positive-unlabeled data as

$$\mathcal{L}_{pu} = \pi_p\mathcal{L}_p(+) + \mathcal{L}_u(-) - \pi_p\mathcal{L}_p(-) \tag{6}$$

where $\mathcal{L}_u(-)$ and $\mathcal{L}_p(-)$ are the expected losses incurred by predicting the unlabeled data with negative and positive ground truth.

Although the above is an unbiased risk estimator, the loss is unbounded as $\mathcal{L}_u(-) - \pi_p\mathcal{L}_p(-)$ can be negative, leading to serious over-fitting under certain circumstances. To alleviate this situation,

a non-negative risk estimator for PU learning is proposed:

$$\mathcal{L}_{pu} = \pi_p \mathcal{L}_p(+) + max\{0, \mathcal{L}_u(-) - \pi_p \mathcal{L}_p(-)\} \tag{7}$$

2.4.2 Raw NN Model

The raw NN models were adopted to set a fundamental baseline. The loss function used was the standard cross-entropy loss.

For Page Blocks data, we fitted a multilayer perceptron (MLP) neural network with 7 fully connected layers. For Fashion MNIST, we fitted a convolutional neural network (CNN) with 3 convolutional layers, 2 pooling layers in between, and 2 fully connected layers with batch normalization. We used the Relu activation function after each layer and log-softmax for output for both NN models.

3 Implementation

During the experiment, the predicted class label is determined as $y = \text{sign}(\Phi(x) - 0.5)$. The value of α is fixed as 0.3, and the value of λ is selected through holdout validation from the set $10^{-4}, 3 \times 10^{-4}, 10^{-3}, \ldots, 1, 3$, unless stated otherwise. We use the Adam optimizer with hyperparameters $(\beta_1, \beta_2) = (0.5, 0.99)$ for VPU.

The performance of VPU is compared with nnPU and raw NN models and the evaluation metrics are on the accuracy, F1-score, recall, and precision of test sets. The mean and standard deviation values are computed from 10 runs.

4 Reproduction Results and Discussion

We compared the performance of our three models on each dataset twice and for the second time, we exchanged the positive and unlabeled data. As discussed in Section 2.2, superscript 1 means class 1 is positive and class 2 is unlabeled and vice versa for superscript 2.

As shown in Figure 1 and Figure 2, this change of label indeed had a significant influence on the model performance as the sizes of the two classes are imbalanced. For Page Blocks, all models achieved very high accuracy. However, VPU was more stable in terms of an F1-score, recall and precision compared with nnPU. Although the performance of the raw NN model was satisfying for Page Blocks, it was much less capable of handling the FashionMNIST image data. It could not capture distinguishing features and tends to assign the same label for the majority of samples, which resulted in its terrible testing scores on FashionMNIST. On the other hand, VPU and

	Accuracy	F1-score	Recall	Precision
VPU - Page Blocks[1]	92.5±0.8	95.7±0.6	95.4±2.1	96.2±1.1
VPU - Page Blocks[2]	94.5±1.6	75.8±4.9	82.9±7.7	70.8±7.0
nnPU - Page Blocks[1]	92.4±0.9	68.2±2.2	79.9±5.5	59.8±3.5
nnPU - Page Blocks[2]	93.3±0.5	96.3±0.3	97.7±0.6	95.0±0.6
Raw - Page Blocks[1]	95.7±1.1	80.9±4.0	88.7±2.3	74.6±6.5
Raw - Page Blocks[2]	96.1±0.6	97.8±0.3	99.0±0.3	96.8±0.9

Figure 1: Test Results on Page Blocks

	Accuracy	F1-score	Recall	Precision
VPU - FMNIST[1]	92.1±0.5	86.1±1.7	81.4±3.9	91.7±2.7
VPU - FMNIST[2]	87.7±2.5	90.1±2.1	87.0±3.8	93.6±1.4
nnPU - FMNIST[1]	92.3±0.3	86.5±0.5	83.3±1.5	90.0±2.0
nnPU - FMNIST[2]	92.0±0.0	94.5±0.5	94.5±0.5	94.5±0.5
Raw - FMNIST[1]	82.1±2.5	58.3±8.0	42.5±8.6	95.1±0.8
Raw - FMNIST[2]	71.3±1.5	22.9±3.6	13.0±2.4	95.7±1.2

Figure 2: Test Results on FashionMNIST

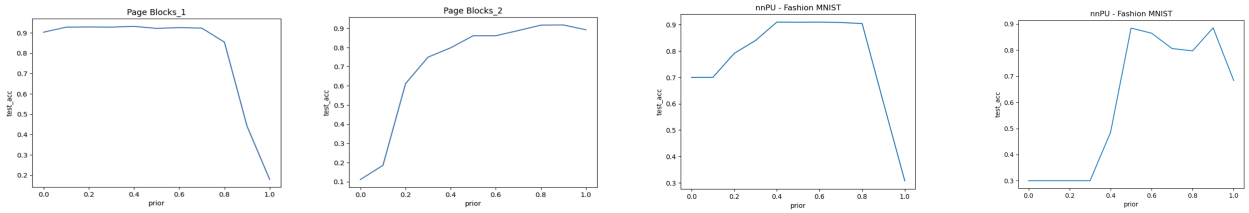

Figure 3: Class Prior vs Test Accuracy (Page Blocks and FashionMNIST)

nnPU had comparable performances on FashionMNIST. However, a proper class prior played an important role in its predicting power. For verification, we experimented with various class prior values from 0 to 1 for both datasets and plotted the accuracy given by nnPU as shown in Figure 3. Its performance was indeed largely affected by the choice of class prior. We conclude that with an inaccurate estimation, nnPU is very likely to lose its strength against VPU.

5 Conclusion

In conclusion, we explored a new approach to deal with positive and unlabeled data, VPU learning, and compared its results with two other baseline models on two datasets. As illustrated by our results, VPU can explore the negative pattern in the PU datasets well. Its performance is more stable compared with the baseline models and has lower requirements on the size of the labelled dataset. While the performance of other PU learning methods such as nnPU may be badly affected by an inaccurate estimate of class prior, VPU is class-prior-free and can be applied to overcome such limitations.

To extend from what VPU has achieved, it could have better performance using advanced techniques developed for measuring differences between distributions for Generative Adversarial Networks (GAN). We may also adopt other variational principles using different statistical distances. Lastly, the input to our models are all real numbers. In the future, we should also consider applying VPU to categorical data based on similar work achieved by other PU learnings.

References

[1] Hui Chen, Fangqing Liu, Yin Wang, Liyue Zhao, and Hao Wu. A variational approach for learning from positive and unlabeled data. *Advances in Neural Information Processing Systems*, 33:14844–14854, 2020.

[2] Marthinus C Du Plessis, Gang Niu, and Masashi Sugiyama. Analysis of learning from positive and unlabeled data. *Advances in neural information processing systems*, 27, 2014.

[3] Charles Elkan and Keith Noto. Learning classifiers from only positive and unlabeled data. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 213–220, 2008.

[4] Ming Hou, Brahim Chaib-Draa, Chao Li, and Qibin Zhao. Generative adversarial positive-unlabelled learning. *arXiv preprint arXiv:1711.08054*, 2017.

[5] Ryuichi Kiryo, Gang Niu, Marthinus C Du Plessis, and Masashi Sugiyama. Positive-unlabeled learning with non-negative risk estimator. *Advances in neural information processing systems*, 30, 2017.

[6] Fuyi Li, Shuangyu Dong, André Leier, Meiya Han, Xudong Guo, Jing Xu, Xiaoyu Wang, Shirui Pan, Cangzhi Jia, Yang Zhang, et al. Positive-unlabeled learning in bioinformatics and computational biology: a brief review. *Briefings in bioinformatics*, 23(1):bbab461, 2022.

[7] Curtis G Northcutt, Tailin Wu, and Isaac L Chuang. Learning with confident examples: Rank pruning for robust classification with noisy labels. *arXiv preprint arXiv:1705.01936*, 2017.

[8] UCI Machine Learning Repository. Page blocks classification data set. `https://archive.ics.uci.edu/ml/datasets/Page+Blocks+Classification`, 1995.

[9] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017.

[10] Chuang Zhang, Dexin Ren, Tongliang Liu, Jian Yang, and Chen Gong. Positive and unlabeled learning with label disambiguation. In *IJCAI*, pages 4250–4256, 2019.