An analytic formulation for positive-unlabeled learning via weighted integral probability metric

PyTorch-based implementation of the paper "An analytic formulation for positive-unlabeled learning via weighted integral probability metric", submitted to ICML 2019.

Directory tree

```
.
├── data.py
├── wmmd.py
├── comparison.py
├── illustration.py
└── main.py
```

- `data.py` generates datasets used for synthetic data analysis in the paper.
- `wmmd.py` contains an implementation of the proposed algorithm.
- `comparison.py` calculates and plots the accuracy and AUC of the proposed algorithm with the synthetic data
- `illustration.py` plots the decision boundary of the proposed classifier with the `two_normal`, `two_circles`, `two_moons` datasets

Requirements

- Python 3
- Pytorch 1.0
- numpy >= 1.8.2
- scipy >= 0.13.3
- sklearn >= 0.19.1
- matplotlib 3.0.2
- tqdm 4.29.1

For GPU configuration: - CUDA 9.0

Quick start

To generate the Figure 1, you simply run the following code.

```
python3 main.py -p 'figure1'
```

After running this code, the result figure is saved as 'figure1.pdf' in current directory. If you want to change the directory or filename, you can use `-d` and `-f` options.

```
python3 main.py -p 'figure1' -d './example_figures/' -f 'my_best_figure.pdf'
```

The available preset plots are:

- Figure1: A plot of accuracy on various class-prior.
- Figure2: A plot of AUC on various class-prior.
- Figure3: An illustration of the WMMD decision boundary with `two_moons` dataset.
- FigureA1: A plot of accuracy on various unlabeled sample size.
- FigureA2: A plot of AUC on various unlabeled sample size.
- FigureA5: An illustration of the WMMD decision boundary with `two_normal` dataset.
- FigureA6: An illustration of the WMMD decision boundary with `two_circles` dataset.

To generate one from Figure A5 ~ Figure A7, you **need type in options for the size of positive and unlabeled samples**. For example, the following code generates a plot of FigureA5 with 10 positive samples and 400 unlabeled samples,

```
python3 main.py -p 'figureA5' -P 10 -U 400
```

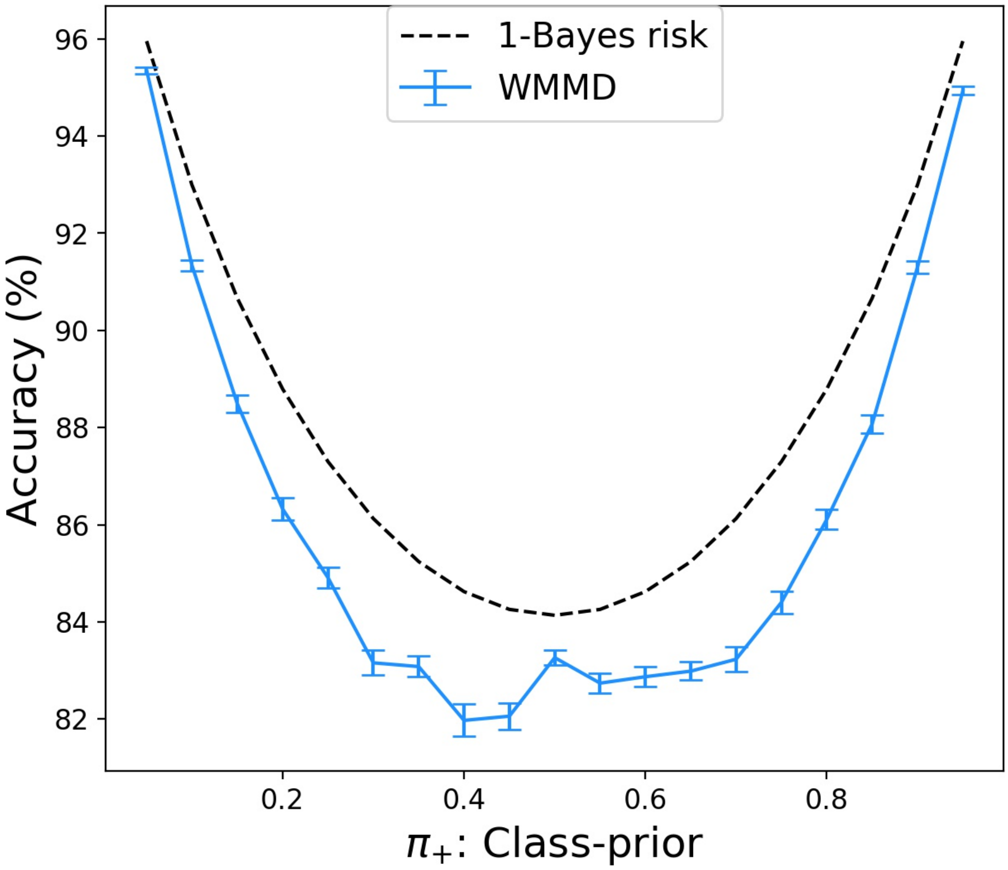
With the `-P` and `-U` options, you can change the positive and unlabeled sample sizes manually.

For the configuration of GPU, add `-g` option with zero-origin GPU ID.

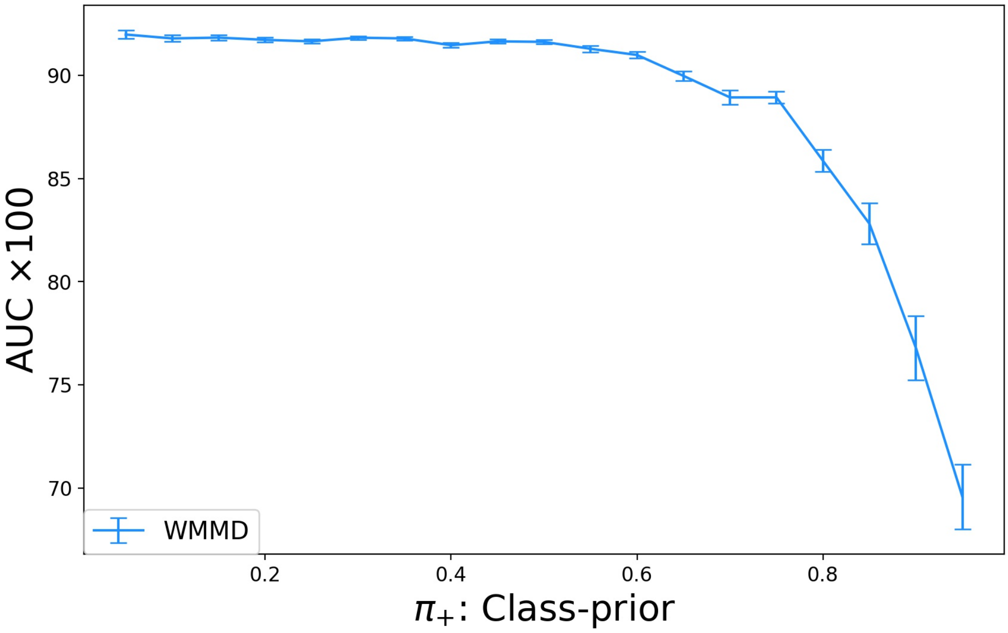
```
python3 main.py -p 'figureA6' -P 10 -U 50 -g 0
```

Example results

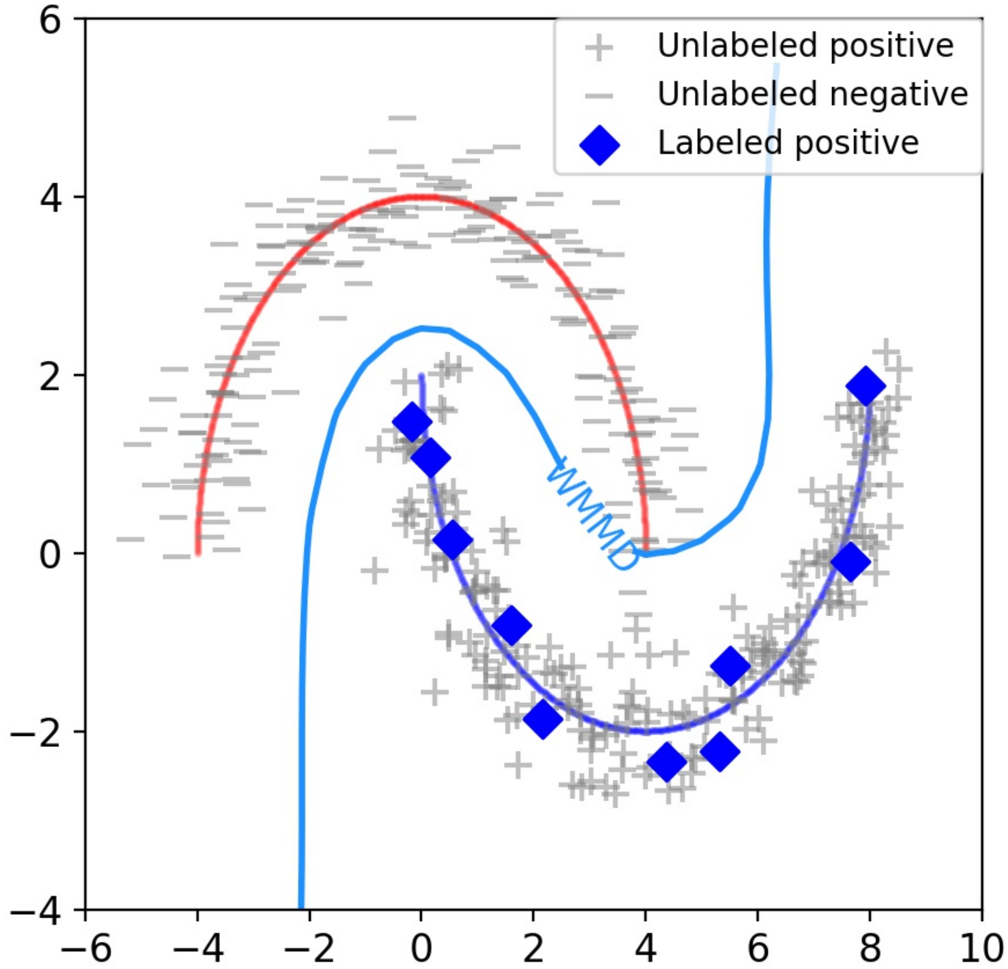
- `figure1.jpg`: a plot of accuracy on various class-prior.



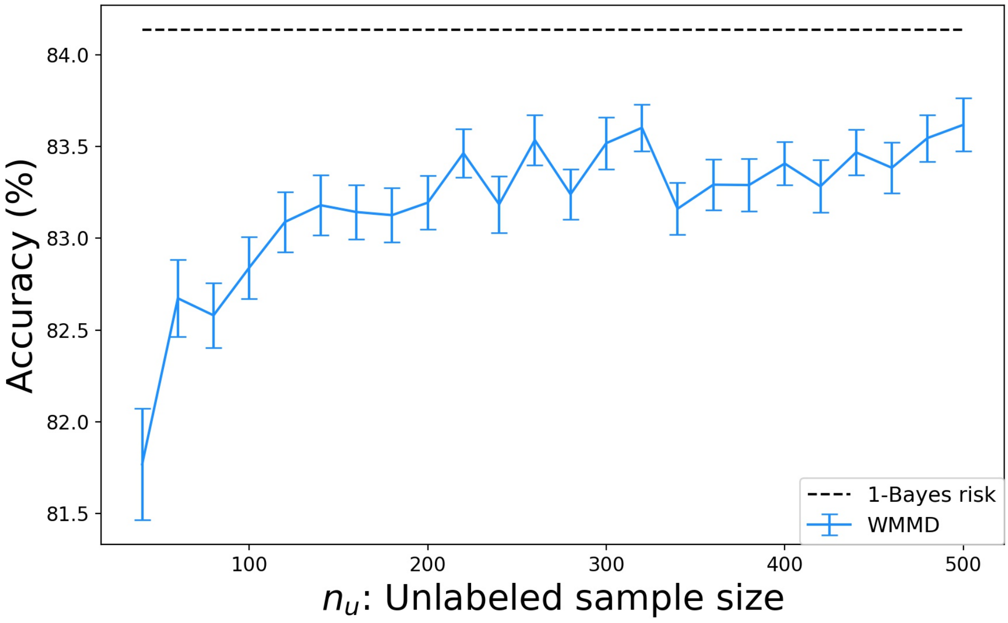
- `figure2.jpg`: a plot of AUC on various class-prior.



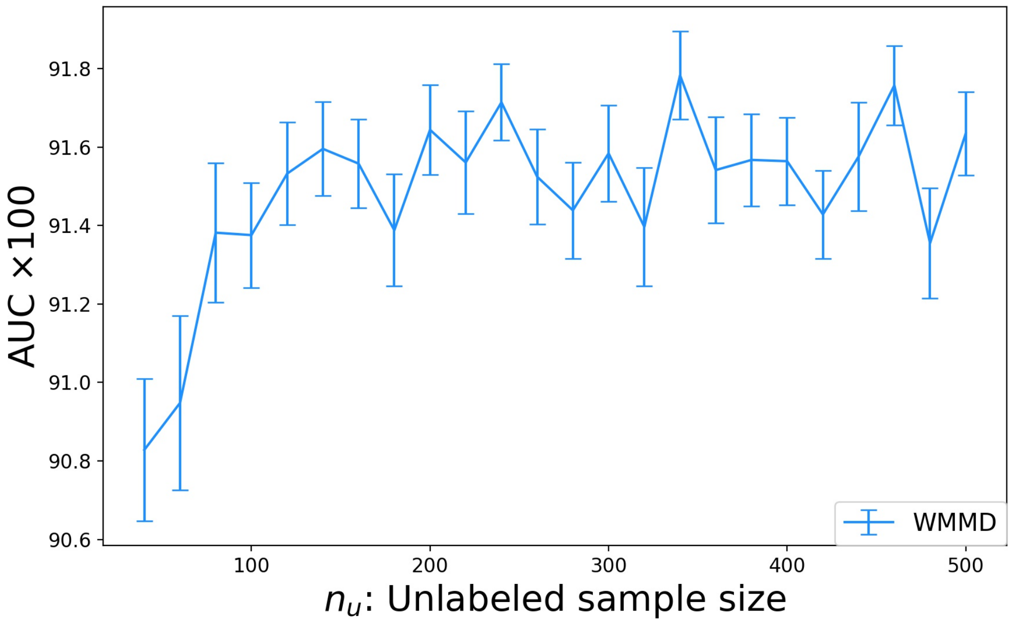
- `figure3.jpg` and `figureA7.jpg`: an illustration of the WMMD decision boundary with `two_moons` dataset.



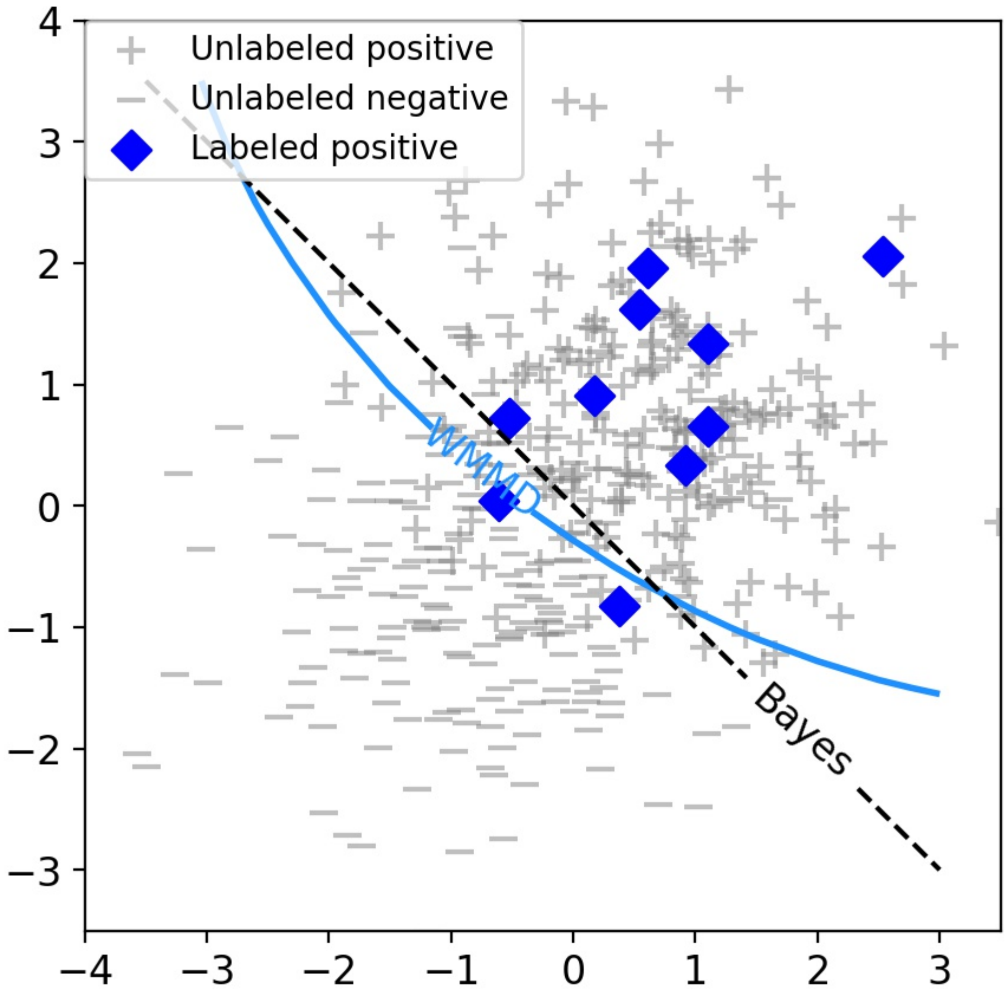
- `figureA1.jpg`: a plot of accuracy on various unlabeled sample size.



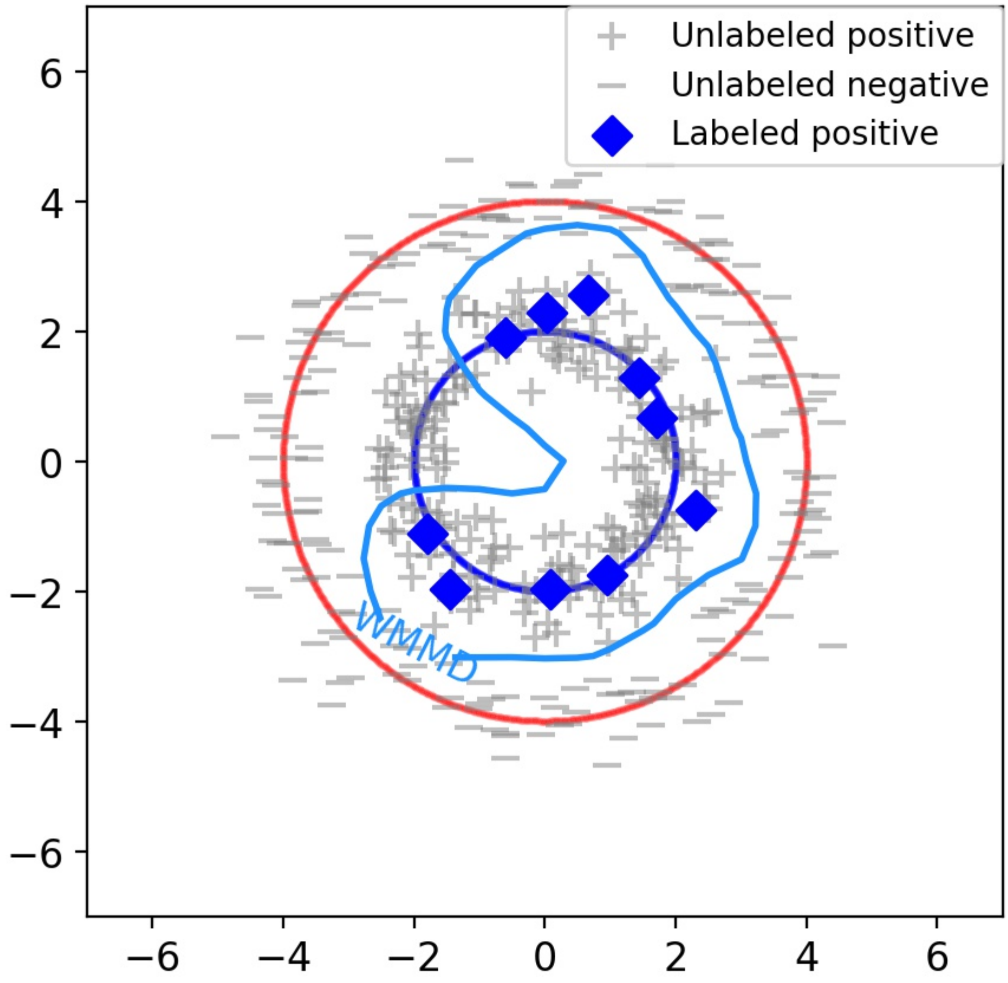
- `figureA2.jpg`: a plot of AUC on various unlabeled sample size.



- `figureA5.jpg`: an illustration of the WMMD decision boundary with `two_normal` dataset.



- `figureA6.jpg`: an illustration of the WMMD decision boundary with `two_circles` dataset.


Unlabeled positive
Unlabeled negative
Labeled positive
WMMD
6
4
2
0
−2
−4
−6
−6
−4
−2
0
2
4
6