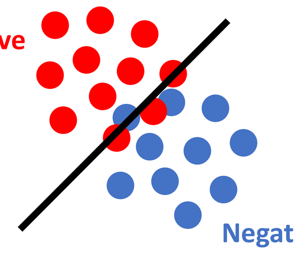
Binary Classification from Positive-Confidence Data

TAKASHI ISHIDA ‖ Univ. of Tokyo & RIKEN   **GANG NIU** ‖ RIKEN   **MASASHI SUGIYAMA** ‖ RIKEN & Univ. of Tokyo

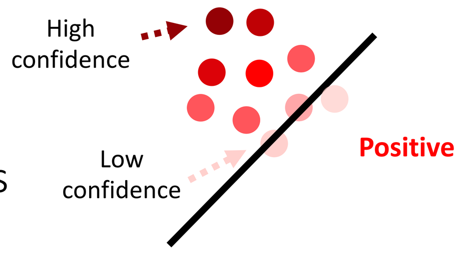
30-Second Summary

- Question: Can we learn a binary classifier from only positive data?
  - Even without *any negative* data or *unlabeled* data?
- Our answer: Yes!
  - If you can equip positive data with confidence (**positive-confidence**), you can successfully learn a binary classifier with optimal convergence rate
- Binary classification from positive-confidence (Pconf) data:
  - Propose a simple empirical risk minimization framework that is,
    - model-independent and optimization-independent
  - Theoretically establish the consistency and an estimation error bound
  - Demonstrate the usefulness through experiments with deep neural nets

Ordinary classification



Pconf classification



Potential Applications

Marketing: Purchase Prediction

- Task: Predict if future customer will purchase your product or rival's product.
- Issue: You only have data of past customers who bought your product (P), and you cannot access rival company's data (N).
- Positive-confidence: You have survey data that asked past customers, how much they wanted to buy your product over rival product. (Normalize it to be probability.)

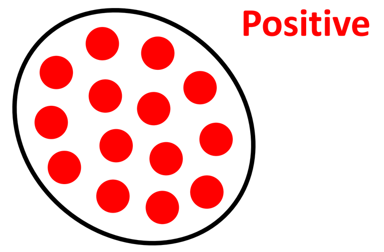
Web Developer: App User Prediction

- Task: Predict if an app user will continue using your app or unsubscribe in the future.
- Issue: Depending on the privacy/opt-out policy or data regulation, the company needs to fully discard the unsubscribed user's data (N). Developers will not have access to users who quit using their services.
- Positive-confidence: Associate a positive-confidence score with each remaining user by, e.g., how actively they use the app. (Normalize it to be probability.)

Related Works

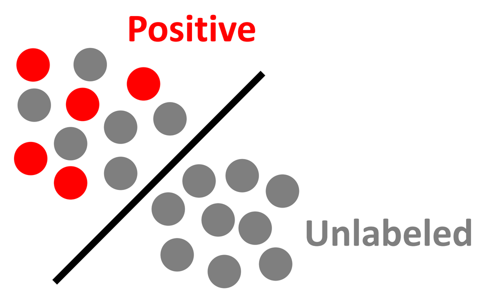
One-class classification (anomaly detection)

- Designed for *describing* the P class (Not for *discriminating* P and N classes!)
- Many one-class methods are motivated:
  - geometrically, by information theory, or by density estimation
- There is no systematic way to tune hyper-parameters to "classify" P and N samples



Positive-Unlabeled (PU) classification

- Uses *additional unlabeled samples* that are sampled from $p(\boldsymbol{x})$
- Directly minimizes the binary classification risk without negative samples
- Requires class prior estimation, which is a difficult task
  - Not necessary in Pconf classification!

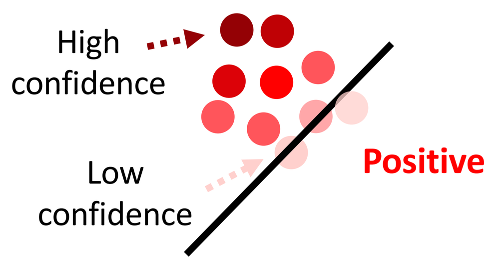


Empirical Risk Minimization Framework

Basic Idea

- Only positive samples → zero information of the negative distribution
  Ex) We don't know the direction of N compared to P distribution
- However, depending on the task, sometimes you can attach a confidence score: *Positive-confidence: 95% dog (5% wolf)*
- Positive-confidence includes the information of the N distribution
  - Will this allow us to learn a good binary classifier?

Pconf classification



Problem Setting

- Goal is to minimize classification risk: $R(g) = \mathbb{E}_{p(\boldsymbol{x},y)}[\ell(yg(\boldsymbol{x}))]$
- We only have pconf data: $\mathcal{X} := \{\boldsymbol{x}_i, r_i\}_{i=1}^n$ ($\mathbb{E}$ is expectation, g is decision function)
  - $\boldsymbol{x}_i$ is positive data drawn from $p(\boldsymbol{x}|y=+1)$
  - r_i is the positive-confidence given by $r_i = p(y=+1|\boldsymbol{x}_i)$
- Issue: We can't directly employ the standard ERM approach!

Empirical Risk Minimization Framework

Theorem

The classification risk can be expressed as

$$R(g) = p(y=+1) \cdot \mathbb{E}_{p(\boldsymbol{x}|y=+1)}\left[\ell(g(\boldsymbol{x})) + \frac{1-r(\boldsymbol{x})}{r(\boldsymbol{x})}\ell(-g(\boldsymbol{x}))\right]$$

if we have $p(y=+1|\boldsymbol{x}) \neq 0$ for all $\boldsymbol{x}$ sampled from $p(\boldsymbol{x})$.

- This means we can ***directly minimize the classification risk*** without access to any negative samples!
- This was ***previously impossible*** with only **hard-labeled** positive samples.
- Intuition: Positive-confidence includes the information of the negative distribution
  - This allow us to discriminate between positive/negative classes

Comparing Proposed and Naïve Methods

Proposed Pconf Method

$$\min_g \sum_{i=1}^n \left[\ell(g(\boldsymbol{x}_i)) + \frac{1-r_i}{r_i}\ell(-g(\boldsymbol{x}_i))\right]$$

Weighted Naïve Method

$$\min_g \sum_{i=1}^n \left[r_i\ell(g(\boldsymbol{x}_i)) + (1-r_i)\ell(-g(\boldsymbol{x}_i))\right]$$

- Naïve weighted method seems more natural and straightforward.
- However it is biased because the population version is not equal to the classification risk.

Theoretical Analysis

For any $\delta > 0$, with probability at least $1-\delta$ (over repeated sampling of data for training $\hat{g}$), we have

$$R(\hat{g}) - R(g^*) \leq 4\pi_+\left(L_\ell + \frac{L_\ell}{C_r}\right)\mathfrak{R}_n(\mathcal{G}) + 2\pi_+\left(C_\ell + \frac{C_\ell}{C_r}\right)\sqrt{\frac{\ln(2/\delta)}{2n}}$$

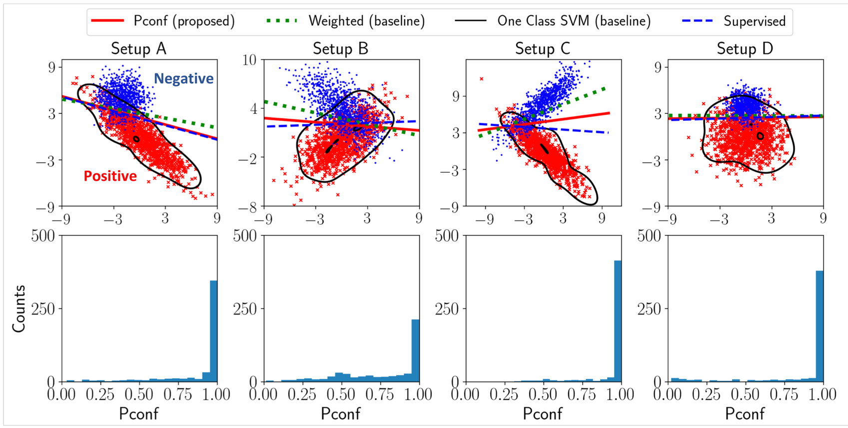
- $\hat{g}$: function that minimizes empirical risk
- g^*: function that minimizes true risk
- $\mathcal{G}$: function set
- L_ℓ: lipshitz constant
- $\mathfrak{R}_n(\mathcal{G})$: Rademacher complexity of $\mathcal{G}$
- π_+: Positive class prior

Experiments

True Positive-Confidence

- Various Gaussian distributions for the positive class and the negative class
- Analytically computed for $p(y=+1|\boldsymbol{x})$ from Gaussian densities and used it as $r(\boldsymbol{x})$
- Linear-in-input model: $g(\boldsymbol{x}) = \boldsymbol{\alpha}^\top \boldsymbol{x} + b$, logistic loss: $\ell_{LL}(z) = \log(1+e^{-z})$

Setup	Pconf	Weighted	Regression	O-SVM	Supervised
A	**89.7 ± 0.6**	88.7 ± 1.2	68.4 ± 6.5	76.0 ± 3.5	89.8 ± 0.7
B	**81.2 ± 1.1**	78.1 ± 1.8	73.2 ± 3.2	71.3 ± 2.3	81.4 ± 1.0
C	**90.2 ± 9.1**	82.7 ± 13.1	50.5 ± 1.7	90.8 ± 1.2	93.6 ± 0.5
D	**91.5 ± 0.5**	90.8 ± 0.7	64.6 ± 5.3	57.1 ± 4.8	91.4 ± 0.5



Noisy Positive-Confidence

- Assuming we know the true positive-confidence exactly is unrealistic in practice
- As noisy positive confidence, we added zero-mean Gaussian noise with standard deviation chosen from {0.01, 0.05, 0.1, 0.2}.
- As the standard deviation gets larger, more noise will be incorporated into positive-confidence.

Setup A

Std.	Pconf	Weighted
0.01	**89.8 ± 0.6**	88.8 ± 0.9
0.05	**89.7 ± 0.6**	88.3 ± 1.1
0.10	**89.2 ± 0.7**	87.6 ± 1.4
0.20	**85.9 ± 2.5**	**85.8 ± 2.5**

Setup C

Std.	Pconf	Weighted
0.01	**92.4 ± 1.7**	84.0 ± 8.2
0.05	**92.2 ± 3.3**	78.5 ± 11.3
0.10	**90.8 ± 9.5**	72.6 ± 12.9
0.20	**88.0 ± 9.5**	65.5 ± 13.1

Setup B

Std.	Pconf	Weighted
0.01	**81.2 ± 0.9**	78.2 ± 1.4
0.05	**80.7 ± 2.3**	78.1 ± 1.4
0.10	**80.8 ± 1.2**	77.8 ± 1.5
0.20	**77.8 ± 1.4**	**77.2 ± 1.9**

Setup D

Std.	Pconf	Weighted
0.01	**91.6 ± 0.5**	90.6 ± 0.9
0.05	**91.5 ± 0.5**	89.9 ± 1.2
0.10	**90.8 ± 0.7**	88.7 ± 1.8
0.20	**87.7 ± 0.8**	85.5 ± 3.7

20 trials, mean and standard deviation of the classification accuracy
If confidence was over 1 or below 0.01, we clipped it to 1 or rounded up to 0.01 respectively.
Best and equivalent methods in red based on 5% t-test, excluding O-SVM & supervised

Benchmark Experiments

- Mean and standard deviation of the classification accuracy over 20 trials for the Fashion-MNIST and CIFAR10 dataset with deep neural networks
- Pconf classification was compared with the baseline Weighted classification method, Auto-Encoder method and fully-supervised method
- Obtained positive-confidence values through a probabilistic classifier trained from a separate set of positive and negative data
- "T-shirt" or "airplane" was chosen as the positive class for Fashion-MNIST and CIFAR10 respectively, and different choices for the negative class
- The best and equivalent methods are shown in red based on the 5% t-test, excluding the Auto-Encoder method and fully-supervised method

P / N	Pconf	Weighted	Auto-Encoder	Supervised
T-shirt / trouser	**92.14 ± 4.06**	85.30 ± 9.07	71.06 ± 1.00	98.98 ± 0.16
T-shirt / pullover	**96.00 ± 0.29**	**96.08 ± 1.05**	70.27 ± 1.22	96.17 ± 0.34
T-shirt / dress	**91.52 ± 1.14**	89.31 ± 1.08	53.82 ± 0.93	96.56 ± 0.34
T-shirt / coat	**98.12 ± 0.33**	**98.13 ± 1.12**	68.74 ± 0.98	98.44 ± 0.13
T-shirt / sandal	**99.55 ± 0.22**	87.83 ± 18.79	82.02 ± 0.49	99.93 ± 0.09
T-shirt / shirt	**83.70 ± 0.46**	**83.60 ± 0.65**	57.76 ± 0.55	85.57 ± 0.69
T-shirt / sneaker	**89.86 ± 13.32**	58.26 ± 14.27	83.70 ± 0.26	100.00 ± 0.00
T-shirt / bag	**97.56 ± 0.99**	95.34 ± 1.00	82.79 ± 0.70	99.02 ± 0.29
T-shirt / ankle boot	**98.84 ± 1.43**	88.87 ± 7.86	85.07 ± 0.37	99.76 ± 0.07

P / N	Pconf	Weighted	Auto-Encoder	Supervised
airplane / automobile	**82.68 ± 1.89**	76.21 ± 2.43	75.13 ± 0.42	93.96 ± 0.58
airplane / bird	**82.23 ± 1.21**	80.66 ± 1.60	54.83 ± 0.39	87.76 ± 4.97
airplane / cat	85.18 ± 1.35	**89.60 ± 0.92**	61.03 ± 0.59	92.90 ± 0.58
airplane / deer	**87.68 ± 1.36**	**87.24 ± 1.58**	55.60 ± 0.53	93.35 ± 0.77
airplane / dog	**89.91 ± 0.85**	**89.08 ± 1.95**	62.64 ± 0.63	94.61 ± 0.45
airplane / frog	**90.80 ± 0.98**	81.84 ± 3.92	62.52 ± 0.68	95.95 ± 0.40
airplane / horse	**89.82 ± 1.07**	85.10 ± 2.61	67.55 ± 0.73	95.65 ± 0.37
airplane / ship	**69.71 ± 2.37**	**70.68 ± 1.45**	52.09 ± 0.42	81.45 ± 8.87
airplane / truck	81.76 ± 2.09	**86.74 ± 0.85**	73.74 ± 0.38	92.10 ± 0.82

YouTube Video: https://youtu.be/2BpJcOf-1XA   Demo code: https://www.github.com/takashiishida/pconf   Paper: https://arxiv.org/abs/1710.07138