http://lamda.nju.edu.cn

A General Formulation for Safely Exploiting Weakly Supervised Data

Lan-Zhe Guo **Yu-Feng Li**

LAMDA Group,
National Key Lab for Novel Software Technology
Nanjing University, China

{guolz, liyf}@lamda.nju.edu.cn

What is this paper about

Weakly supervised data is one important machine learning data

It suffers one serious issue

- The usage of weakly supervised data may even degenerate performance, which means, it could be outperformed by its supervised counterpart using only a small number of labeled data

Contribution of this work

In this work, we consider to learn a safe prediction for weakly supervised learning, where safe means it will not be worse than its supervised counterpart. We propose a general formulation and give theoretical analysis. The experiments also show quite promising results

Outline

- **Introduction**
- Proposed Approach
- Experiments
- Conclusion

What is Weakly Supervised Learning

- Weakly supervised learning use the data that does not require a large amount of precise label information
- For Example:
  - Label Noise Learning [Fr´enay and Verleysen 2014]
  - Semi-Supervised Learning [Chapelle et al. 2006)]
  - Domain Adaptation [Pan and Yang 2010]
  - ...

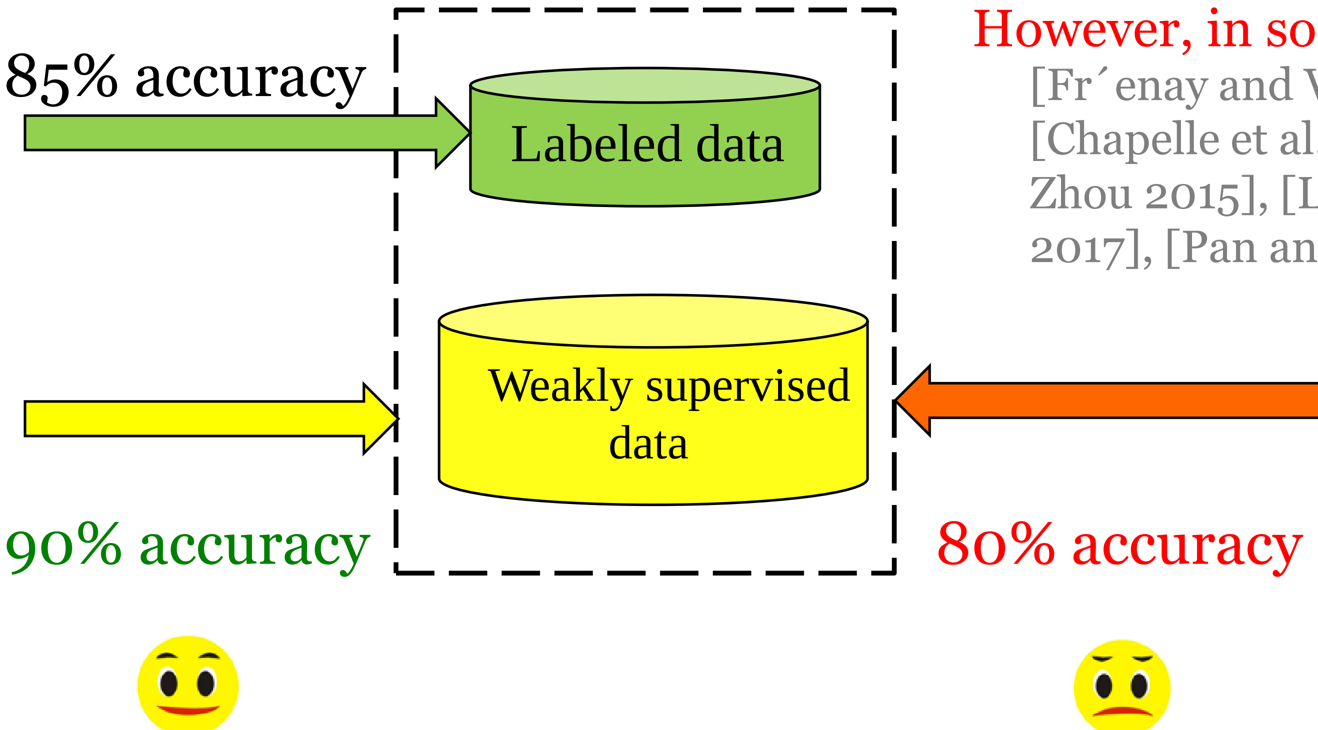
Examples

- Label Noise Learning
  - We have only a small number of high-quality labeled data and a lot of noisy labeled data
- Semi-Supervised Learning
  - We have only limited labeled data and need to leverage a number of unlabeled data
- Domain Adaptation
  - Label information in target domain is not sufficient and we need to exploit further label information from other domains

Weakly supervised learning is not safe

LAMDA Learning And Mining from DatA http://lamda.nju.edu.cn

It is often expected that weakly supervised data can help improve performance since more data are used. However, it sometimes fails.



However, in some cases

[Fr´enay and Verleysen 2014], [Chapelle et al. 2006], [Li and Zhou 2015], [Li, Zha and Zhou 2017], [Pan and Yang 2010]

Outline

- Introduction
- **Proposed Approach**
- Experiments
- Conclusion

The Basic Setup

- Suppose we have a set of weakly supervised learning predictions $\{y_i\}_{i=1}^{n}$
- These base predictions can be obtained in various ways, e.g., by different type of algorithms
- Moreover, we can easily train a supervised method with the use of only limited labeled data and let y_0 denote the prediction

The goal: to learn a safe prediction $g((y_1, \cdots y_n), y_0)$, which often outperform, and will not be worse than y_0

A Direct Approach

Learning And Mining from DatA
http://lamda.nju.edu.cn

Suppose we know the ground-truth $\boldsymbol{y}^*$, we can directly maximize the performance gain

$$\max_{\mathbf{y} \in \mathbb{H}^u} \ell(\mathbf{y}_0, \mathbf{y}^*) - \ell(\mathbf{y}, \mathbf{y}^*)$$

Trained on high-quality labeled data only

Ground Truth

However

Obviously, y^* is unknown

We can construct y^* with $\{y_i\}_{i=1}^{n}$, and consider the worst case for the requirement of safeness.

SafeW

$$\max_{\mathbf{y}\in\mathbb{H}^u} \boxed{\min_{\boldsymbol{\alpha}\in\mathcal{M}}} \ell(\mathbf{y}_0, \sum_{i=1}^{n} \alpha_i \mathbf{y}_i) - \ell(\mathbf{y}, \sum_{i=1}^{n} \alpha_i \mathbf{y}_i)$$

Worst case consideration [Li and Zhou, ICML2011/TPAMI2015; Balsubramani and Freund, COLT2015]

Three Questions about the formulation

LAMDA
Learning And Mining from DatA
http://lamda.nju.edu.cn

$$\max_{\mathbf{y}\in\mathbb{H}^u}\min_{\boldsymbol{\alpha}\in\mathcal{M}} \ell(\mathbf{y}_0,\sum_{i=1}^{n}\alpha_i\mathbf{y}_i)-\ell(\mathbf{y},\sum_{i=1}^{n}\alpha_i\mathbf{y}_i)$$

- Is this formulation reasonable?
- How to setup the set of weights $\mathcal{M}$?
- How to solve it efficiently?

Is this formulation reasonable?

LAMDA Learning And Mining from DatA http://lamda.nju.edu.cn

$$\max_{\mathbf{y}\in\mathbb{H}^u}\min_{\boldsymbol{\alpha}\in\mathcal{M}} \ell(\mathbf{y}_0, \sum_{i=1}^{n}\alpha_i\mathbf{y}_i) - \ell(\mathbf{y}, \sum_{i=1}^{n}\alpha_i\mathbf{y}_i)$$

Theoretical analysis:

Theorem 1. *Suppose the ground-truth* $\mathbf{y}^*$ *can be constructed by the base learners, i.e.,* $\mathbf{y}^* \in \{\mathbf{y} | \sum_{i=1}^{b}\alpha_i\mathbf{y}_i, \boldsymbol{\alpha} \in \mathcal{M}\}$. *Let* $\hat{\mathbf{y}}$ *and* $\hat{\boldsymbol{\alpha}}$ *be the optimal solution to Eq.(1), we then have* $\ell(\hat{\mathbf{y}}, \mathbf{y}^*) \leq \ell(\mathbf{y}_0, \mathbf{y}^*)$ *and* $\hat{\mathbf{y}}$ *has already achieved the maximal performance gain against* $\mathbf{y}_0$.

If the assumption is not satisfied?

Learning And Mining from DatA
http://lamda.nju.edu.cn

If $\ell(\cdot,\cdot)$ is η-Lipschitz, i.e., $|\ell(y_1, y_2) - \ell(y_1, y_3| \leq \eta||y_2 - y_3||_1$

Let $\boldsymbol{\beta}^* = \underset{\boldsymbol{\beta} \in \mathcal{M}}{\arg\min}\ \ell(\sum_{i=1}^{n} \beta_i \mathbf{y}_i, \mathbf{y}^*)$ and $\boldsymbol{\epsilon} = \mathbf{y}^* - \sum_{i=1}^{n} \beta_i^* \mathbf{y}_i$

We have,

Theorem 4. *The performance gain of $\hat{\mathbf{y}}$ against $\mathbf{y}_0$, i.e., $\ell(\mathbf{y}_0, \mathbf{y}^*) - \ell(\hat{\mathbf{y}}, \mathbf{y}^*)$, has a lower-bound $-2\eta||\boldsymbol{\epsilon}||_1$.*

Setup $\mathcal{M}$

For regression $C_{ij} = (y_i - u_i)^\top (y_j - u_j)$, for classification $C_{ij} = y_i^\top y_j$

We prove that $\mathbf{C\alpha}$ indicates the performance of the base learner

And we can think the base learner has a lower-bound performance [Balsubramani and Freund 2015]

Hence, we can setup $\mathcal{M} = \{\boldsymbol{\alpha} | \mathbf{C}\boldsymbol{\alpha} \geq \boldsymbol{\delta}, \mathbf{1}^\top \boldsymbol{\alpha} = 1, \boldsymbol{\alpha} \geq \mathbf{0}\}$

Moreover, if we have prior knowledge, we can setup $\mathcal{M}$ more flexible

Optimization

Original Form:

$$\max_{\mathbf{y}\in\mathbb{H}^u} \min_{\boldsymbol{\alpha}\in\mathcal{M}} \ell(\mathbf{y}_0, \sum_{i=1}^{n} \alpha_i \mathbf{y}_i) - \ell(\mathbf{y}, \sum_{i=1}^{n} \alpha_i \mathbf{y}_i)$$

Usually non-convex and not easy to solve

For regression task, we can get a convex optimization: $\min_{\boldsymbol{\alpha}\in\mathcal{M}} \ell(\mathbf{y}_0, \sum_{i=1}^{n} \alpha_i \mathbf{y}_i)$

For classification task and hinge loss, we can get a linear programming:

$$\min_{\boldsymbol{\alpha}\in\mathcal{M}} \ell(\mathbf{y}_0, \sum_{i=1}^{n} \alpha_i \mathbf{y}_i) + \frac{1}{u} \| \sum_{i=1}^{n} \alpha_i \mathbf{y}_i \|_1 - 1$$

Become much easier to solve

Outline

- Introduction
- Proposed Approach
- **Experiments**
- Conclusion

Label Noise Learning

❑Setup

- 8 frequently-used datasets
- 30% high-quality labeled data and 70% noisy data which their labels are random reversed with a probability p%.

❑Compared Methods

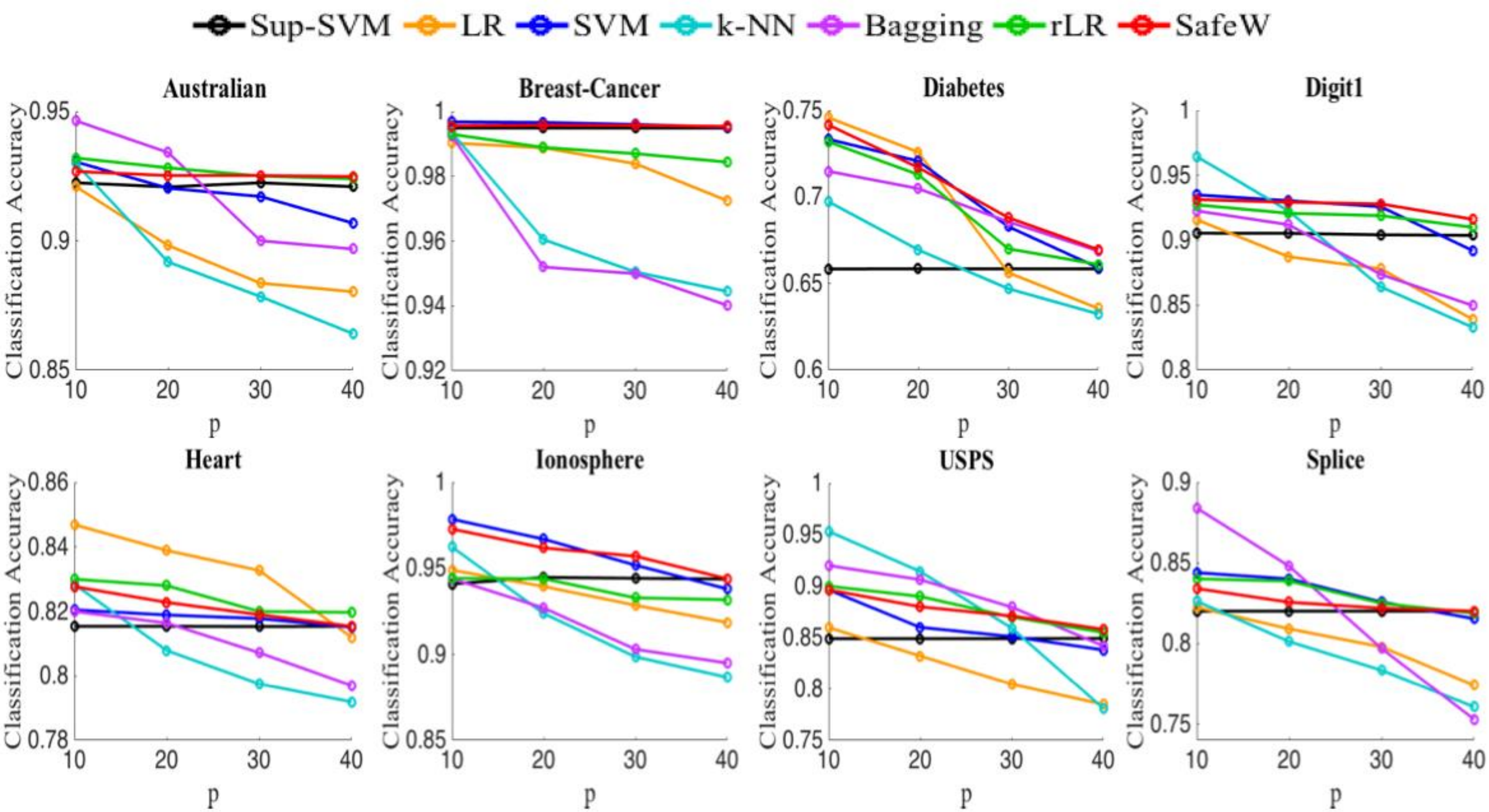
- Baseline
  - Sup-SVM
- 2 state-of-art noisy robust methods
  - Bagging [Fr´enay and Verleysen 2014],
  - rLR [Bootkrajang and Kab´an 2012],
- 3 traditional methods
  - SVM
  - k-NN
  - Logistic Regression(LR)

❑SafeW: adopted SVM, k-NN and LR as base learners.

Label Noise Learning

Learning And Mining from DatA
http://lamda.nju.edu.cn

Classification accuracy on 8 datasets with p range from 10% to 40%



Domain Adaptation

- Setup
  - 2 benchmark datasets: 20newgroup, landmine
- Compared Methods
  - Baseline
    - Logistic Regression
  - 4 transfer learning methods
    - Original
    - MIDA [Yan, Kou, and Zhang 2016]
    - TCA [Pan et al. 2011]
    - TrAdaBoost [Dai et al. 2007]
- SafeW: adopted Original, MIDA and TCA as base learners.

Domain Adaptation

20newsgroup: 19,997 news and 20 groups

Dataset	Logistic Regression	Original	MIDA	TCA	TrAdaBoost	SafeW
Comp vs Rec	$.7028 \pm .0091$	$\mathbf{.7492 \pm .0135}$	$\mathbf{.7961 \pm .0197}$	$\mathbf{.7940 \pm .0162}$	$\mathbf{.8077 \pm .0155}$	$\mathbf{.7956 \pm .0170}$
Comp vs Sci	$.8225 \pm .0662$	$\boxed{.7985 \pm .0194}$	$\mathbf{.8946 \pm .0188}$	$.8255 \pm .0172$	$\mathbf{.8583 \pm .0201}$	$\mathbf{.8925 \pm .0212}$
Comp vs Talk	$.8423 \pm .0685$	$\boxed{.8022 \pm .0182}$	$\boxed{.8231 \pm .0164}$	$.8434 \pm .0110$	$\boxed{.8247 \pm .0143}$	$.8451 \pm .0158$
Sci vs Talk	$.7294 \pm .1045$	$.7100 \pm .0121$	$\mathbf{.7456 \pm .0164}$	$\boxed{.7022 \pm .0092}$	$.7166 \pm .0213$	$\mathbf{.7468 \pm .0153}$
Rec vs Sci	$.8006 \pm .0758$	$\boxed{.7754 \pm .0161}$	$.8033 \pm .0151$	$\mathbf{.8440 \pm .0118}$	$.8016 \pm .0151$	$\mathbf{.8435 \pm .0157}$
Rec vs Talk	$.8278 \pm .0446$	$.8276 \pm .0115$	$\mathbf{.8566 \pm .0105}$	$\mathbf{.8580 \pm .0128}$	$\mathbf{.8415 \pm .0113}$	$\mathbf{.8579 \pm .0105}$
Average	.7876	.7805	.8199	.8112	.8084	.8302
Win/Tie/Loss against LR		1/2/3	4/1/1	3/2/1	3/2/1	**5/1/0**

Directly using weakly supervised data often degenerate performance while SafeW does not suffer this problem.

Domain Adaptation

Landmine: 29 domain, 9 features

Dataset	Logistic Regression	Original	MIDA	TCA	TrAdaBoost	SafeW
Domain 20	.9215 ± .0173	.9237 ± .0034	**.9265 ± .0039**	.9255 ± .0045	.9183 ± .0029	**.9271 ± .0035**
Domain 21	.9360 ± .0095	.9310 ± .0047	.9384 ± .0045	.9304 ± .0051	.9261 ± .0033	.9396 ± .0038
Domain 22	.9594 ± .0051	.9555 ± .0038	.9506 ± .0065	**.9650 ± .0017**	.9095 ± .0026	**.9648 ± .0016**
Domain 23	.9361 ± .0095	.9310 ± .0041	**.9424 ± .0045**	.9314 ± .0051	**.9627 ± .0043**	**.9426 ± .0038**
Domain 24	.9535 ± .0052	.9524 ± .0029	.9447 ± .0025	.9432 ± .0029	.9535 ± .0034	.9550 ± .0024
Average	.9413	.9387	.9405	.9391	.9340	.9458
Win/Tie/Loss against LR		0/3/2	2/1/2	1/1/3	1/2/2	**3/2/0**

We have similar observations.

Semi-Supervised Learning

❑Setup

- 8 commonly used regression datasets
- 10 labeled data are chosen for each dataset

❑Compared Methods

- Baseline
  - 1NN
- Semi-Supervised Regressor
  - Self-kNN [Yarowsky 1995]
  - Self-LS [Hastie, Tibshirani, and Friedman 2001]
- Ensemble methods
  - Average
  - Safer [Li, Zha and Zhou 2017]

❑For Average, Safer and SafeW: adopted Self-kNN(Euclidean), Self-kNN(Cosine) and Self-LS as base learners.

Semi-Supervised Learning

Mean Square Error on 8 datasets

Dataset	1NN	Self-*k*NN(Euclidean)	Self-*k*NN(Cosine)	Self-LS	Average	Safer	SAFEW
abalone	.020 ± .010	**.014 ± .005**	**.014 ± .003**	**.013 ± .004**	**.012 ± .003**	**.013 ± .005**	**.013 ± .005**
bodyfat	.019 ± .005	.018 ± .006	.019 ± .005	.041 ± .013	.023 ± .009	.018 ± .007	.017 ± .005
cadata	.083 ± .029	**.063 ± .012**	**.058 ± .009**	**.056 ± .007**	**.057 ± .009**	**.060 ± .013**	**.057 ± .005**
cpusmall	.024 ± .012	.027 ± .011	.028 ± .009	.025 ± .010	.024 ± .005	.025 ± .011	.024 ± .009
housing	.039 ± .010	.036 ± .009	.033 ± .006	.036 ± .009	**.034 ± .008**	**.034 ± .009**	**.033 ± .005**
mg	.051 ± .009	**.039 ± .006**	**.038 ± .006**	**.035 ± .015**	**.038 ± .014**	**.038 ± .006**	**.038 ± .006**
mpg	.022 ± .007	.020 ± .006	.018 ± .006	.021 ± .008	.020 ± .006	**.019 ± .004**	**.018 ± .004**
pyrim	.023 ± .006	**.021 ± .005**	.022 ± .005	.052 ± .014	**.020 ± .007**	**.020 ± .006**	**.020 ± .006**
Ave. Mse.	.035	.030	.029	.035	.029	.030	.028
Win/Tie/Loss against 1NN		4/3/1	3/4/1	3/3/2	5/2/1	**6/2/0**	**6/2/0**

SafeW also obtain safe predictions

Outline

- Introduction
- Proposed Approach
- Experiments
- **Conclusion**

Conclusion

- We propose a general formulation for safely exploiting weakly supervised data

- It has three advantages

  - Has safeness guarantee for commonly used loss functions in both regression and classification tasks

  - Can setup the weight of base learner flexibly

  - Can be solved globally in an efficient manner

Thanks!