Self-Paced Robust Learning for Leveraging Clean Labels in Noisy Data

Xuchao Zhang,[1] Xian Wu,[2] Fanglan Chen,[1] Liang Zhao,[3] Chang-Tien Lu[1]
[1]Discovery Analytics Center, Virginia Tech, Falls Church, VA,
[2]University of Notre Dame, Notre Dame, IN,
[3]George Mason University, Fairfax, VA
[1]{xuczhang, fanglanc, ctlu}@vt.edu, [2]xwu9@nd.edu, [3]lzhao9@gmu.edu

AAAI 2020

Motivation

□ Phenomena

1. Real-world datasets contain erroneously labeled data samples.
2. Well-labeled data is usually expensive.

□ Problem

How to train a robust model by using large-scale noisy data in conjunction with a small set of clean data ?

Problem Formulation

- Split the samples

1. Clean Set (a small set of well-labeled samples with little data corruption) :

$$\mathcal{D}_s = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_k, y_k)\}$$

2. Noisy Set (a weakly labeled dataset) :

$$n \gg k$$

$$\mathcal{D}_w = \{(\boldsymbol{x}_{k+1}, y_{k+1}), \ldots, (\boldsymbol{x}_n, y_n)\}$$

- Goal

$$\hat{\boldsymbol{w}} = \arg\min_{\boldsymbol{w} \in \mathcal{R}^p} \sum_{i \in \mathcal{D}_s \cup \mathcal{D}_w^+} \mathcal{L}(y_i, f(\boldsymbol{x}_i, \boldsymbol{w})) + \psi(\boldsymbol{w})$$

the uncorrupted data samples in $\mathcal{D}_w$

- Challenges

1. $\mathcal{D}_w^+$ in $\mathcal{D}_w$ is unknown. Can't simply ignore $\mathcal{D}_w$, because $n \gg k$.
2. $\mathcal{D}_w$ can be extremely noisy.

SPRL Algorithm

Objective Function

the total loss of $\mathcal{D}_s$

$$\underset{\boldsymbol{w}\in\mathcal{R}^n,\boldsymbol{v}\in[0,1]}{\arg\min}\ \mathcal{J}(\boldsymbol{w},\boldsymbol{v};\lambda)=\sum_{i=1}^{k}\mathcal{L}(y_i,f(\boldsymbol{x}_i,\boldsymbol{w}))+\sum_{i=k+1}^{n}v_i\mathcal{L}(y_i,f(\boldsymbol{x}_i,\boldsymbol{w}))+\|\boldsymbol{w}\|_2^2+\theta\|\boldsymbol{w}-\tilde{\boldsymbol{w}}\|_2^2-\lambda\sum_{i=k+1}^{n}v_i\,,\quad(2)$$

the total loss of $\mathcal{D}_w$

$$\tilde{\boldsymbol{w}}=\underset{\boldsymbol{w}}{\arg\min}\sum_{i=1}^{k}\mathcal{L}(y_i,f(\boldsymbol{x}_i,\boldsymbol{w}))+\psi(\boldsymbol{w})$$

Control the difference between the estimated model and the model trained in the clean set only.

Table 1: Math Notations

Notations	Explanations
p	number of feature in data matrix X
k	number of samples in the clean dataset
n	number of samples in the entire dataset
$X, \boldsymbol{y}$	data matrix and its corresponding label vector
$\boldsymbol{w}$	parameters of estimated model
$\tilde{\boldsymbol{w}}$	model parameter trained in the clean set
$\boldsymbol{v}$	instance weight vector $v_i \in [0,1]$
λ	parameter to control the learning pace
μ	step size of parameter λ
$\mathcal{L}$	loss function of estimated model

SPRL Algorithm

- fix $\boldsymbol{w}$

$$v_i^{t+1} = \arg\min_{v_i \in [0,1]} \sum_{i=k+1}^{n} v_i \mathcal{L}(y_i, f(\boldsymbol{x}_i, \boldsymbol{w}^t)) - \lambda^t \sum_{i=k+1}^{n} v_i$$

closed-form solution:

$$v_i^{t+1} = \begin{cases} 1, & \text{if } \mathcal{L}(y_i, f(\boldsymbol{x}_i, \boldsymbol{w}^t)) < \lambda^t \\ 0, & \text{otherwise} \end{cases}$$

- fix $\boldsymbol{v}$

$$\boldsymbol{w}^{t+1} = \arg\min_{\boldsymbol{w} \in \mathcal{R}^p} \sum_{i=1}^{k} \mathcal{L}(y_i, f(\boldsymbol{x}_i, \boldsymbol{w}))$$

$$+ \sum_{i=k+1}^{n} v_i^{t+1} \mathcal{L}(y_i, f(\boldsymbol{x}_i, \boldsymbol{w})) + \|\boldsymbol{w}\|_2^2 + \theta \|\boldsymbol{w} - \tilde{\boldsymbol{w}}\|_2^2$$

Algorithm 1: SPRL ALGORITHM

Input: $X \in \mathcal{R}^{p \times n}$, $\boldsymbol{y} \in \mathcal{R}^n$, $\theta \in \mathcal{R}$, $\lambda^0 \in \mathcal{R}$, $\lambda_\infty \in \mathcal{R}$, $\mu \in \mathcal{R}$
Output: solution $\boldsymbol{w}^{(t+1)}$, $\boldsymbol{v}^{(t+1)}$

```
w̃ ← arg min_w Σ_{i=1}^k L(y_i, f(x_i, w)) + ψ(w)
Initialize w^0 = w̃, ε > 0, t ← 0
repeat
    for i = k+1 ... n do
        v_i^{t+1} ← ∞(L(y_i, f(x_i, w^t)) < λ^t)
    Update w^{t+1} by Equation (6) with fixed v^{t+1} and w̃.
    λ^{t+1} ← λ^t * μ
    if λ^{t+1} > λ_∞ then
        λ^{t+1} ← λ_∞
    t ← t + 1
until ||J(w^{t+1}, v^{t+1}; λ^{t+1}) − J(w^t, v^t; λ^t)||_2 < ε
return w^{t+1}, v^{t+1}
```

Control the size of training set.

Convergence Analysis

☐ Assumption 1 (Lower Bound)

The loss function $\mathcal{L}$ in problem (2) has a lower bound $\mathcal{B}$ as follows:

$$\mathcal{B} = \min_{\boldsymbol{w}} \mathcal{L}\big(y, f(\boldsymbol{x}, \boldsymbol{w})\big) > -\infty$$

E.g. least-squares loss、hinge loss: $\mathcal{B} = 0$

☐ Lemma 1

The objective function $\mathcal{J}$ in Equation (2) is lower bounded as follows:

$$\lim_{t\to\infty} \mathcal{J}(\boldsymbol{w}^t, \boldsymbol{v}^t; \lambda^t) > -\infty$$

☐ Theorem 1

When Assumption 1 is satisfied, Algorithm 1 converges with the following property:

$$\lim_{t\to\infty} \left\|\mathcal{J}^{t+1} - \mathcal{J}^t\right\|_2 = 0$$

Convergence Analysis

Proof Lemma 1 :

$$\mathcal{J}(\boldsymbol{w}^t, \boldsymbol{v}^t; \lambda^t) \overset{(a)}{\geq} \sum_{i=1}^{k} \mathcal{B} + \sum_{i=k+1}^{n} v_i^t \mathcal{B} + \|\boldsymbol{w}^t\|_2^2 + \theta\|\boldsymbol{w}^t - \tilde{\boldsymbol{w}}\|_2^2 - \lambda^t \sum_{i=k+1}^{n} v_i^t \overset{(b)}{\geq} k\mathcal{B} + \sum_{i=k+1}^{n} v_i^t \mathcal{B} - (n-k) \cdot \lambda_\infty$$

inequality (a) $\mathcal{L}(y_i, f(\boldsymbol{x}_i, \boldsymbol{w}^t)) \geq \mathcal{B}$ and $v_i^t \geq 0$

inequality (b) $\|\boldsymbol{w}^t\|_2^2 \geq 0$, $\theta\|\boldsymbol{w}^t - \tilde{\boldsymbol{w}}\|_2^2 \geq 0$

$\because$ $\lambda \leq \lambda_\infty$ and $v_i \in [0,1]$

$\therefore$ $\lambda^t \sum_{i=k+1}^{n} v_i^t \leq (n-k) \cdot \lambda_\infty$

$\because$ when $B \geq 0$ we have $\sum_{i=k+1}^{n} v_i^t \mathcal{B} \geq 0$

when $B < 0$ we have $\sum_{i=k+1}^{n} v_i^t \mathcal{B} \geq (n-k) \cdot \mathcal{B}$

$\therefore$ $\mathcal{J}(\boldsymbol{w}^t, \boldsymbol{v}^t; \lambda^t) \geq k\mathcal{B} + \min\{0, (n-k) \cdot \mathcal{B}\} - (n-k) \cdot \lambda_\infty = k\mathcal{B} + (n-k) \cdot (\min\{0, \cdot\mathcal{B}\} - \lambda_\infty)$

$\because$ $\mathcal{B} > -\infty$ and λ_∞ is constant

$\therefore$ $\mathcal{J}(\boldsymbol{w}^t, \boldsymbol{v}^t; \lambda^t) > -\infty$ for $\forall t = 1 \ldots \infty$

Convergence Analysis

Proof Theorem 1 :

(1) $\mathcal{J}$ is monotonically decreased.

$$\mathcal{J}(\boldsymbol{w}^{t+1},\boldsymbol{v}^{t+1};\lambda^{t+1})\overset{(a)}{\leq}\sum_{i=1}^{k}\mathcal{L}(y_i,f(\boldsymbol{x}_i,\boldsymbol{w}^{t+1}))+\sum_{i=k+1}^{n}v_i^{t+1}\mathcal{L}(y_i,f(\boldsymbol{x}_i,\boldsymbol{w}^{t+1}))+\|\boldsymbol{w}^{t+1}\|_2^2+\theta\|\boldsymbol{w}^{t+1}-\tilde{\boldsymbol{w}}\|_2^2-\lambda^t\sum_{i=k+1}^{n}v_i^{t+1}$$

inequality (a) : $\because$ λ increases monotonically. $\therefore$ $\lambda^{t+1}\geq\lambda^t$ and $v_i^t\geq 0$

$\because$ Line 7 in Algorithm 1

$$\therefore \sum_{i=k+1}^{n}v_i^{t+1}\mathcal{L}(y_i,f(\boldsymbol{x}_i,\boldsymbol{w}^{t+1}))-\lambda^t\sum_{i=k+1}^{n}v_i^{t+1}\leq\sum_{i=k+1}^{n}v_i^{t}\mathcal{L}(y_i,f(\boldsymbol{x}_i,\boldsymbol{w}^{t+1}))-\lambda^t\sum_{i=k+1}^{n}v_i^{t}$$

$$\therefore \mathcal{J}(\boldsymbol{w}^{t+1},\boldsymbol{v}^{t+1};\lambda^{t+1})\leq\sum_{i=1}^{k}\mathcal{L}(y_i,f(\boldsymbol{x}_i,\boldsymbol{w}^{t+1}))+\sum_{i=k+1}^{n}v_i^{t}\mathcal{L}(y_i,f(\boldsymbol{x}_i,\boldsymbol{w}^{t+1}))+\|\boldsymbol{w}^{t+1}\|_2^2+\theta\|\boldsymbol{w}^{t+1}-\tilde{\boldsymbol{w}}\|_2^2-\lambda^t\sum_{i=k+1}^{n}v_i^{t}$$

$\because$ Line 5 in Algorithm 1

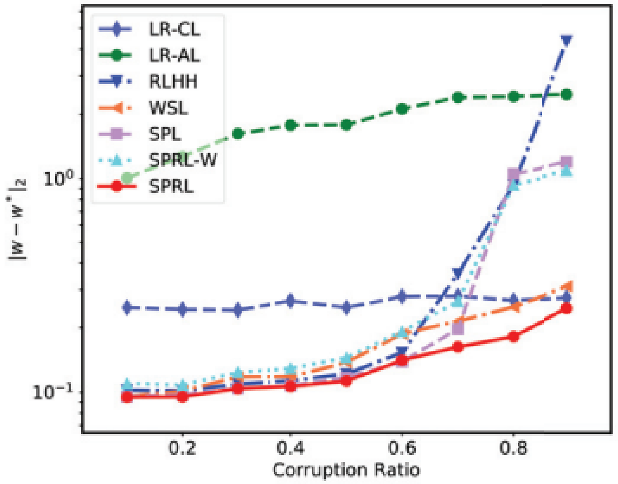
$$\therefore \mathcal{J}(\boldsymbol{w}^{t+1},\boldsymbol{v}^{t+1};\lambda^{t+1})\leq\sum_{i=1}^{k}\mathcal{L}(y_i,f(\boldsymbol{x}_i,\boldsymbol{w}^{t}))+\sum_{i=k+1}^{n}v_i^{t}\mathcal{L}(y_i,f(\boldsymbol{x}_i,\boldsymbol{w}^{t}))+\|\boldsymbol{w}^{t}\|_2^2+\theta\|\boldsymbol{w}^{t}-\tilde{\boldsymbol{w}}\|_2^2-\lambda^t\sum_{i=k+1}^{n}v_i^{t}=\mathcal{J}(\boldsymbol{w}^{t},\boldsymbol{v}^{t};\lambda^{t})$$

(2) $\because$ $\mathcal{J}$ is monotonically decreased and it has a lower bound.

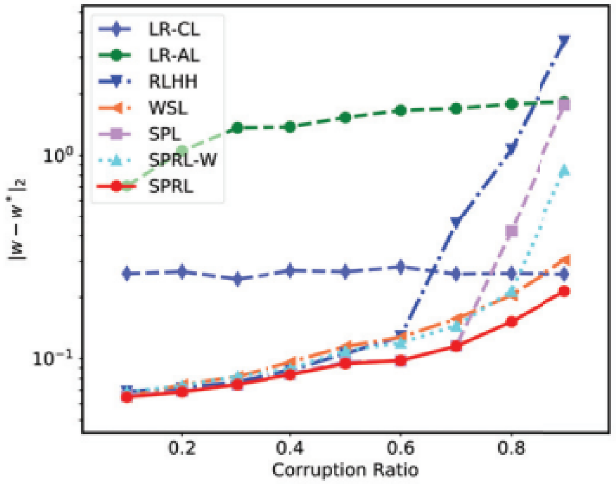
$\therefore$ $\|\mathcal{J}^{t+1}-\mathcal{J}^t\|_2<\varepsilon$ for $\forall\varepsilon>0$

Experiments: Regression

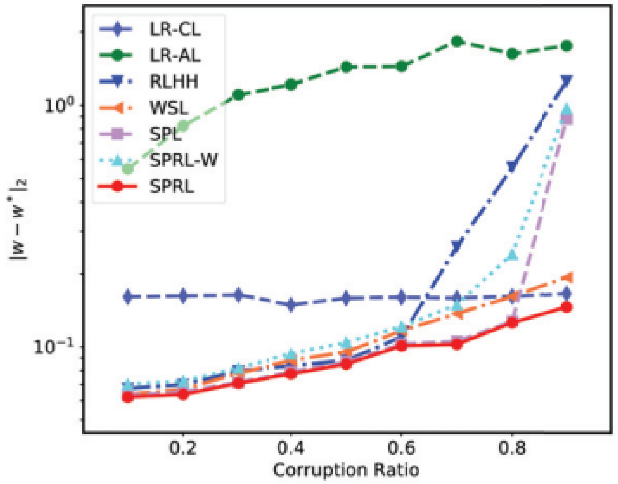
$$y_* = X^T w_* + \varepsilon \qquad y = y_* + u$$



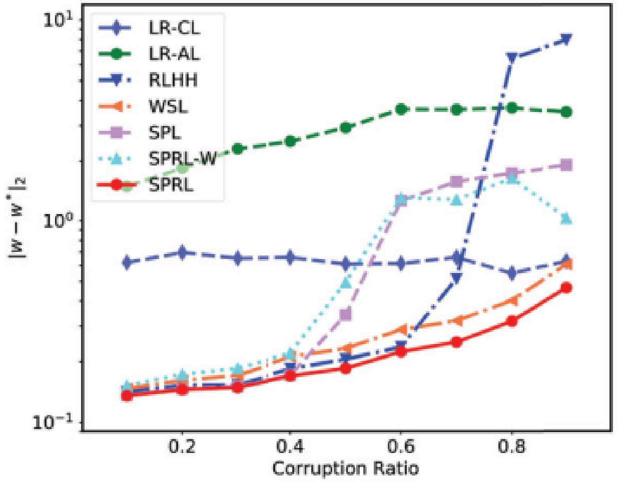
(a) p=400, k=1K, n=6K, dense noise



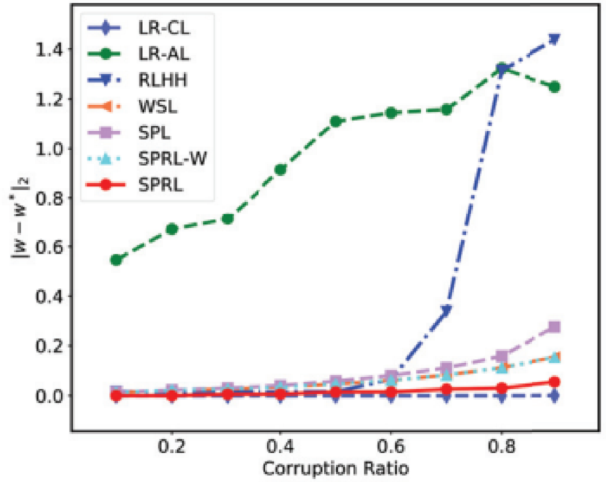
(b) p=400, k=1K, n=11K, dense noise



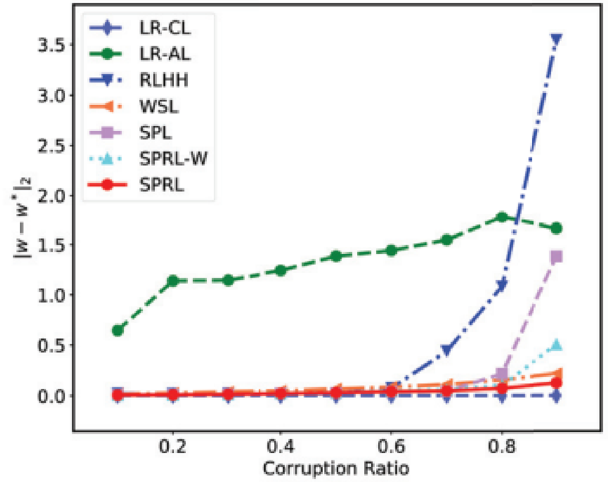
(c) p=400, k=2K, n=12K, dense noise



(d) p=800, k=1K, n=6K, dense noise



(e) p=200, k=1K, n=11K, no dense noise



(f) p=400, k=1K, n=11K, no dense noise

Figure 1: Performance on Regression Coefficient Recovery for Different Corruption Ratios in Uniform Distribution.

Experiments: Regression

Table 2: Mean Absolute Error of Blog Feedback Prediction

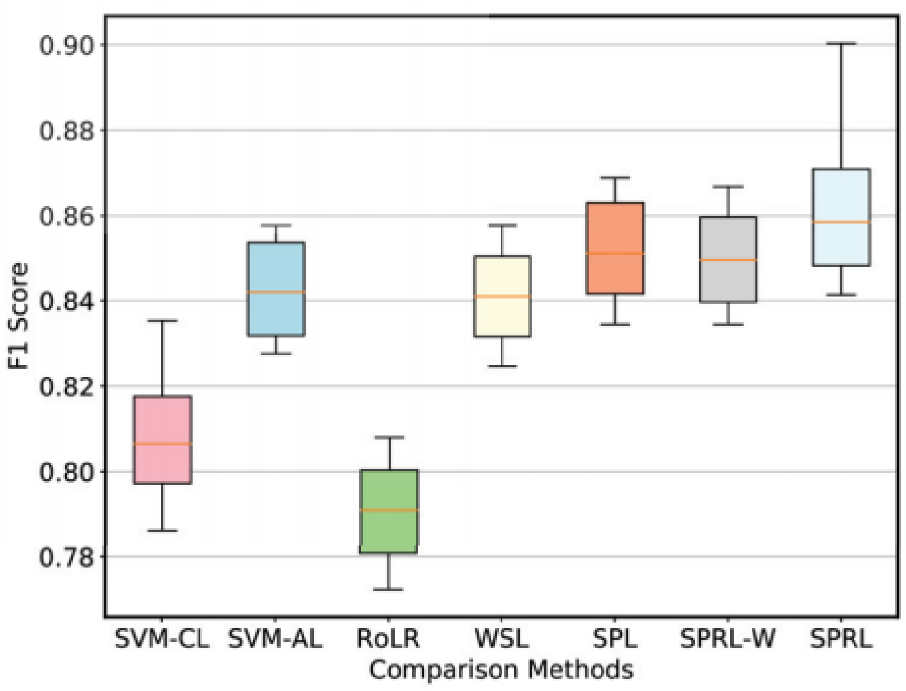
	Corruption Ratio					
	10%	**30%**	**50%**	**70%**	**90%**	**Avg.**
LR-CL	1.159	1.161	1.153	1.164	1.173	1.162
LR-AL	7.254	17.116	10.459	17.226	8.334	12.0778
WSL	0.981	1.280	2.562	2.154	1.375	1.6704
SPL	0.973	1.189	3.666	4.382	4.525	2.947
SPRL-W	**0.919**	2.627	2.493	4.547	5.797	3.2766
SPRL	0.971	**1.107**	**1.036**	**1.053**	**1.046**	**1.0426**

Experiments: Binary Classification

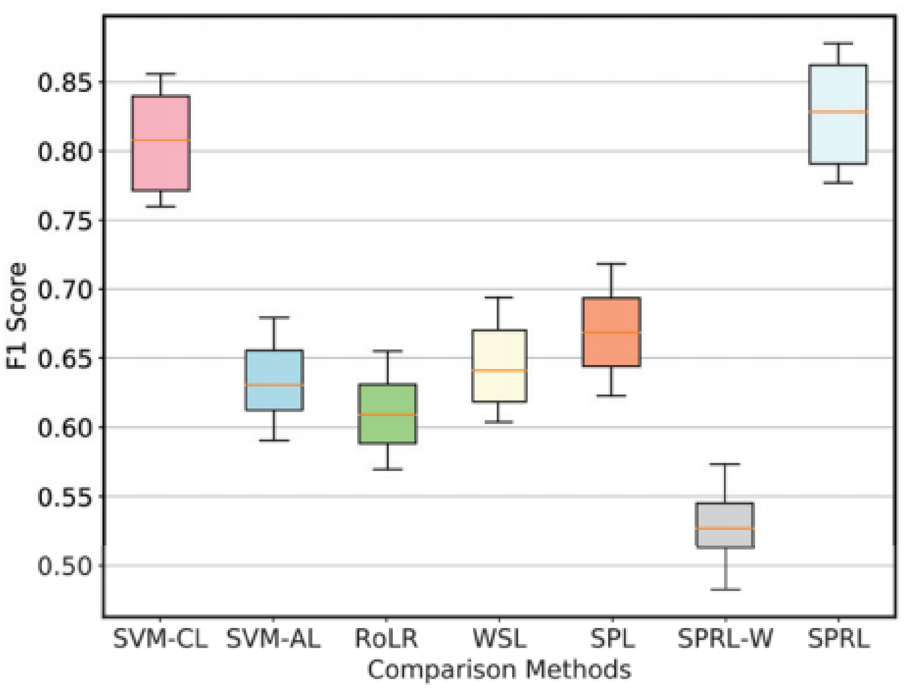
Table 3: Performance on Binary Classification (F1, Precision, Recall)

	feature=200, clean set=100, noisy set=5K				feature=400, clean set=100, noisy set=5K			
	10%	20%	30%	40%	10%	20%	30%	40%
SVM-CL	0.657,0.656,0.659	0.654,0.650,0.658	0.676,0.667,0.686	0.674,0.688,0.661	0.628,0.629,0.628	0.639,0.640,0.638	0.626,0.621,0.630	0.620,0.627,0.613
SVM-AL	0.928,0.927,0.929	0.900,0.902,0.898	0.835,0.831,0.838	0.750,**0.754**,0.747	0.918,0.916,0.920	0.860,0.861,0.859	**0.786,0.796**,0.776	0.665,0.670,0.661
RoLR	0.814,0.817,0.810	0.842,0.840,0.845	0.804,0.795,0.814	0.724,0.730,0.719	0.827,0.834,0.820	0.788,0.790,0.785	0.747,0.758,0.736	0.650,0.659,0.641
WSL	0.886,0.889,0.883	0.791,0.792,0.789	0.745,0.739,0.752	0.706,0.715,0.697	0.870,0.873,0.868	0.786,0.789,0.783	0.690,0.690,0.690	0.644,0.653,0.635
SPL	0.946,0.946,0.946	0.903,0.905,0.902	0.809,0.805,0.813	0.665,0.666,0.665	0.916,0.921,0.912	0.824,0.822,0.826	0.739,0.744,0.735	0.608,0.614,0.602
SPRL-W	0.944,0.942,0.946	0.913,0.916,0.910	0.799,0.796,0.802	0.694,0.699,0.689	0.905,0.903,0.906	0.811,0.815,0.808	0.754,0.760,0.749	0.637,0.647,0.628
SPRL	**0.968,0.965,0.971**	**0.922,0.918,0.928**	**0.871,0.874,0.866**	**0.751**,0.742,**0.754**	**0.935,0.936,0.932**	**0.864,0.863,0.865**	0.780,0.785,**0.783**	**0.681,0.691,0.674**
	feature=200, clean set=200, noisy set=5K				**feature=200, clean set=200, noisy set=10K**			
	10%	20%	30%	40%	10%	20%	30%	40%
SVM-CL	0.758,0.756,0.759	0.722,0.720,0.725	0.734,0.730,0.739	0.734,0.738,0.730	0.715,0.718,0.712	0.730,0.734,0.726	0.732,0.728,0.736	0.701,0.697,0.705
SVM-AL	0.942,0.939,0.944	0.897,0.891,0.904	0.853,0.846,0.861	0.749,0.743,0.756	0.948,0.946,0.950	0.932,0.934,0.930	0.898,0.899,0.897	0.787,0.790,0.784
RoLR	0.833,0.833,0.834	0.834,0.834,0.834	0.808,0.806,0.811	0.699,0.693,0.705	0.879,0.877,0.882	0.886,0.884,0.888	0.665,0.668,0.662	0.771,0.770,0.771
WSL	0.905,0.899,0.911	0.827,0.825,0.829	0.796,0.794,0.798	0.743,0.747,0.740	0.902,0.900,0.904	0.856,0.861,0.851	0.798,0.801,0.795	0.722,0.718,0.727
SPL	0.950,0.951,0.949	0.905,0.899,0.912	0.810,0.810,0.810	0.665,0.665,0.665	0.967,0.965,0.969	0.959,0.963,0.954	0.869,0.875,0.864	0.687,0.689,0.686
SPRL-W	0.949,0.949,0.949	0.896,0.892,0.900	0.822,0.823,0.821	0.745,0.736,0.755	0.966,0.964,0.969	0.950,0.953,0.946	0.902,0.903,0.900	0.721,0.722,0.721
SPRL	**0.963,0.967,0.960**	**0.926,0.925,0.927**	**0.876,0.878,0.875**	**0.768,0.780,0.763**	**0.981,0.977,0.985**	**0.959,0.954,0.963**	**0.920,0.928,0.912**	**0.787,0.782,0.793**

Experiments: Binary Classification



(a) Clean set=2K, Noisy set=10K, Corruption Ratio=10%



(b) Clean set=2K, Noisy set=10K, Corruption Ratio=50%

Figure 2: Sentiment Classification Performance on Movie Reviews

Thanks