Safe Semi-Supervised Learning

Yu-Feng Li (李宇峰)
National Key Laboratory for Novel Software Technology, Nanjing University, China
URL: http://lamda.nju.edu.cn/liyf/
Email: liyf@nju.edu.cn

Joint work with Zhi-Hua Zhou (Nanjing University), James Kwok (HKUST), Ivor Tsang (UTS)
In order to have a good generalization performance, supervised learning methods often assume that a large amount of labeled data are available.
Labeled Data Is Expensive

- However, labeling process is **expensive** in many real tasks
  - Disease diagnosis
  - Drug detection
  - Image classification
  - Text categorization
  - ...

Require human efforts and material resources
Exploiting Unlabeled Data

- Collection of unlabeled data is usually cheaper

- Two popular schemes that exploit unlabeled data to help improve the performance of supervised learning
  - **Semi-supervised learning**: the learner tries to exploit the unlabeled examples by itself.
  - **Active learning**: the learner actively selects some unlabeled examples to query from an oracle
**Semi-Supervised Learning**

- **Surveys and Books**
SSL Applications

• Many applications

• Text Categorization [Joachims 1999; Joachims, 2002]

• Email Classification [Kockelkorn et al., 2003]

• Image Retrieval [Wang et al., 2003]

• Bioinformatics [Kasabov & Pang, 2004]

• Named Entity Recognition [Goutte et al., 2002]
Four Popular SSL Paradigms

- **Generative models** [B.M Shahshahani & D.A. Landgrebe, TGRS94; D.J. Miller & H.S. Uyar, NIPS96; etc.]

- **Disagreement-based methods** [Blum & Mitchell, ICML98; Balcan et al., NIPS05; Zhou & Li, TKDE10; etc.]

- **Graph-based methods** [Blum & Chawla, ICML01; Zhu et al., ICML03; Zhou et al., NIPS05; Belkin et al., JMLR06; etc.]

- **Semi-Supervised SVMs** [Vapnik, STL98; Bennett & Demiriz, NIPS99; Joachims, ICML99; Chapelle & Zien, ICML05; etc.]
Generative Methods

- Assume that the labeled and unlabeled data is generated from a joint distribution. After that, it estimates distribution parameters as well as a label assignment of unlabeled data so that the likelihood is maximized.

- Different kinds of generative models have been used, e.g.,
  - Mixture of Gaussians [B.M Shahshahani & D.A. Landgrebe, TGRS94]
  - Mixture of Experts [D.J. Miller & H.S. Uyar, NIPS96]
  - Naïve Bayes [K. Nigam et al., MLJ00]

- Expectation-Maximization (EM) algorithm is often employed to estimate the parameters and the label assignment
Disagreement-based Methods

- Train multiple learners to exploit the unlabeled data, and then utilize the ‘disagreement’ information among the learners to help improve the performance.

- Various disagreement-based methods have been used, e.g.,
  - Co-training: exploit two views to derive two learners and show that if two views are sufficient and redundant, Co-training can be boosted to arbitrary high accuracy [Blum & Mitchell, ICML98]
  - Tri-training: three learners are employed to improve the generalization [Zhou & Li, TKDE10]

The seminal work of co-training [Blum & Mitchell, ICML98] won the ‘10-year best paper’ award in ICML’08.
Graph-based Methods

- Construct a weighted graph on the labeled and unlabeled training examples
  - The edge weights correspond to some relationship (such as similarity/distance) between the samples
- Assume that examples connected with heavy edge tend to have the same label
- Infer a label assignment of unlabeled data so that the label inconsistency w.r.t. graph is minimized.
- Different kinds of inference algorithms have been developed.

Semi-Supervised SVMs (S3VMs)

In [Vapnik, SLT’98], it is shown that large margin could help improve the generalization learning bound.

Large-margin separator
(or, low-density separator)

Theorem 1 ([Vapnik, 1998])
Consider hyperplanes $h(\mathbf{x}) = \text{sign}(\mathbf{x} \cdot \mathbf{w} + b)$ as hypothesis space $H$. If the attribute vectors of a training sample (2) and a test sample (3) are contained in a ball of diameter $D$, then there are at most

$$N_r < \exp\left(d \left(\frac{n + k}{d} + 1\right)\right), d = \min\left(a, \left[\frac{D^2}{\rho^2}\right] + 1\right)$$

equivalence classes which contain a separating hyperplane with

$$\forall_{i=1}^n \left| \frac{\mathbf{w}}{||\mathbf{w}||} \cdot \mathbf{x}_i + b \right| \geq \rho \quad \forall_{j=1}^k \left| \frac{\mathbf{w}}{||\mathbf{w}||} \cdot \mathbf{x}_j^* + b \right| \geq \rho$$

(i.e. margin larger or equal to $\rho$). $a$ is the dimensionality of the space, and $[b]$ is the integer part of $b$. 

http://lamda.nju.edu.cn
S3VMs: Formulation

\[
\min_{\hat{y} \in \mathcal{B}} \min_{\mathbf{w}} \Omega(\mathbf{w}) + C_1 \sum_{i=1}^{l} \ell(\mathbf{w}, \mathbf{x}_i, y_i) + C_2 \sum_{j=l+1}^{N} \ell(\mathbf{w}, \mathbf{x}_j, \hat{y}_j)
\]

SVM

Optimize a large-margin label assignment w.r.t. some prior constraints for possible label assignments, e.g., the label proportion of unlabeled data is similar to that of labeled data.

The seminal work of S3VM [Joachims, ICML99] won the ‘10-year best paper’ award in ICML’09.
Challenges

Large-scale data
[AIStats09; ECML09; IEEE TIT13; etc.]

Real-time requirement
[ICML09; NIPS 12; SDM16; etc.]

Performance guarantee
This talk

Avoid suffering serious mistake
[AAAI10/13/16; etc.]
SSL Revisit

Previous SSL assumes that unlabeled data will help improve the performance. **This however, may be not hold.**

However, in some cases [Cozman et al., ICML03] [Balcan et al. ICMLworkshop05] [Jebara et al. ICML09][Zhang & Oles, ICML00][Wang et al., CVPR03] [Chapelle et al., ICML06]…

SSL is not safe, i.e., the exploitation of unlabeled data may hurt the performance. Such phenomena undoubtedly affect the deployment of SSL in real tasks.
Discussions in literature

- **Generative method:** [Cozman et al., 2003] conjectured that the performance degeneration is caused by incorrect model assumption. However, it is very difficult to make a correct model assumption without sufficient domain knowledge.

- **Co-training method:** Incorrect pseudo-labels may mislead the learning process. One possible solution is to employ data editing process [Li and Zhou, 2005]. However, it only works for dense data.

- **Graph-based method:** Graph construction is the crucial problem. However, how to develop a good graph in general situations remains an open problem.
Discussions in literature

- **S3VMs**: The correctness of S3VMs has been studied on very small data sets [Chapelle et al., 2008]. However, it is unclear whether S3VM is safe for regular and large scale data sets.

- There are also some general discussions from a theoretical perspective [Balcan and Blum, 2010; Ben-David et al., 2008; Singh et al., 2009].

- To our best knowledge, few safe SSL approaches have been proposed.

How to develop safe SSL methods which do not significantly reduce the performance?
Outline

• Improve the quality of optimization solution
  • WELL SVM [Li et al., JMLR13]

• Address the uncertainty of model selection
  • S4VM [Li and Zhou, TPAMI15]

• Overcome the variety of performance measures
  • UMVP [Li et al., AAAI16]
Outline

• Improve the quality of optimization solution
  • WELL SVM [Li et al., JMLR13]

• Address the uncertainty of model selection
  • S4 VM [Li and Zhou, TPAMI15]

• Overcome the variety of performance measures
  • UMVP [Li et al., AAAI16]
S3VM Optimization

- Revisit the optimization of S3VM

\[
\min_{\hat{y} \in \mathcal{B}} \min_w \Omega(w) + C_1 \sum_{i=1}^{l} \ell(w, x_i, y_i) + C_2 \sum_{j=l+1}^{N} \ell(w, x_j, \hat{y}_j)
\]

- The optimization involves many poor properties
  - Mixed integer programming
  - Non-convex
  - Many local minima

A poor quality of optimization solution affects the effectiveness of S3VM

http://lamda.nju.edu.cn
Previous Efforts

- Global optimization algorithms, e.g.,
  - Branch-and-Bound [Chepelle et al., NIPS06]
  - Deterministic Annealing [Sindhwani et al., ICML06]
  - Continuation Method [Chepelle et al., ICML06]

- Good thing: **good performance on very small data sets**
- Weakness: **poor scalability** (i.e., could not handle with more than **several hundred examples**)

http://lamda.nju.edu.cn
Previous Efforts

- Local optimization algorithm, e.g.,
  - Local Combinatorial Search [Joachims, ICML99]
  - Alternating Optimization [Zhang et al., ICML09]
  - Constrained Convex-Concave Procedure (CCCP) [Collobert et al., JMLR06]

- Good thing: **Good scalability**
- Weakness: easy get stuck in local minima, suffer from suboptimal performance
Previous Efforts

- SDP convex relaxation [Xu et al., 2005; De Bie and Cristianini, 2006]
  - Relax S3VMs as convex Semi-Definite Programming (SDP)
  - SDP typically scales $O(n^{6.5})$ where $n$ is the sample size [Zhang et al., TNN2011].

- Good thing: promising performance
- Weakness: poor scalability (i.e., could not handle with more than several thousand examples)

Previous solutions either suffer from scalability issue or local optima problem

Can we have a scalable and promising solution?

Yes, we propose a WellSVM approach
Intuition

Do not know the label of unlabeled data

Hard, Not Scalable

Given a label assignment for unlabeled data

Easy, Scalable
The basic idea is to generate a set of informative label assignments and then learn an optimal combination of these label assignments so that margin is maximized.

Since the optimization procedure does not involve integer variables, it becomes easy and scalable.
Formal Derivation

- **S3VMs Primal and its Duality**

\[
\min_{\hat{y} \in B} \min_{w} \Omega(w) + C_1 \sum_{i=1}^{l} \ell(w, x_i, y_i) + C_2 \sum_{j=l+1}^{N} \ell(w, x_j, \hat{y}_j)
\]

**Duality:** A minimax problem

\[
\min_{\hat{y} \in B} \max_{\alpha \in A} \quad G(\alpha, \hat{y}) := 1'\alpha - \frac{1}{2}\alpha'(K \odot \hat{y}\hat{y}')\alpha.
\]

- **Minimax Relaxation**

**WellSVM**

\[
\max_{\alpha \in A} \min_{\hat{y} \in B} \quad G(\alpha, \hat{y}) := 1'\alpha - \frac{1}{2}\alpha'(K \odot \hat{y}\hat{y}')\alpha.
\]

**S3VMs**

\[
\min_{\hat{y} \in B} \max_{\alpha \in A} \quad G(\alpha, \hat{y}) := 1'\alpha - \frac{1}{2}\alpha'(K \odot \hat{y}\hat{y}')\alpha.
\]
Relaxation

- Rewritten as

\[
\max_{\alpha \in \mathcal{A}} \left\{ \max_\theta \theta \right\}
\]

s.t. \( G(\alpha, \hat{y}_t) \geq \theta, \ \forall \hat{y}_t \in \mathcal{B} \),

WellSVM is a convex relaxation of S3VMs

\[ \min_{\mu \in \mathcal{M}} \max_{\alpha \in \mathcal{A}} \sum_{t: \hat{y}_t \in \mathcal{B}} \mu_t G(\alpha, \hat{y}_t), \]

where \( \mu \) is the vector of \( \mu_t \)'s, \( \mathcal{M} \) is the simplex \{ \( \mu \mid \sum_t \mu_t = 1, \mu_t \geq 0 \} \), and \( \hat{y}_t \in \mathcal{B} \).

Proposition 1. The objective of WELL SVM can be rewritten as the following optimization problem:
Optimization

$$\max_{\alpha \in \mathcal{A}} \left\{ \max_{\theta} \theta \right\}$$

s.t. \( G(\alpha, \hat{y}_t) \geq \theta, \ \forall \hat{y}_t \in \mathcal{B} \),

- exponential number of constraints, direct optimization computationally intractable
- Typically not all these constraints are active at optimality
- Including only a subset of them: a very good approximation

**Cutting-Plane** method
- Generate a violated label assignment

$$y^* = \arg\max_{\hat{y} \in \mathcal{B}} \hat{y}'(K \odot \alpha \alpha') \hat{y}. \quad \text{Can be solved by sorting.}$$
Optimization

\[
\max_{\alpha \in A} \left\{ \max_\theta \ \theta \ \mid \ G(\alpha, \hat{y}_t) \geq \theta, \ \forall \hat{y}_t \in B \right\},
\]

- exponential number of constraints, direct optimization computationally intractable
- Typically not all these constraints are active at optimality
- Including only a subset of them: a very good approximation
- Cutting-Plane method
  - Optimal combination

\[
\min_{\mu \in M} \max_{\alpha \in A} \ 1'\alpha - \frac{1}{2} \alpha' \left( \sum_{t=1}^{T} \mu_t K \circ \hat{y}_t \hat{y}_t' \right) \alpha,
\]

Multiple Kernel Learning, can be solved by state-of-the-art SVM software, which is scalable
Convergence Analysis

• $-G(\alpha, y)$ is $\lambda$-strongly convex and $M$-Lipschitz.
• Let $p^{(t)}$ be the optimal objective value of at the $t$-th iteration.

\[ p^{(t+1)} \leq p^{(t)} - \eta, \text{ where } \eta = \left( -c + \sqrt{c^2 + 4\epsilon} \right)^2, \text{ and } c = M \sqrt{2/\lambda}. \]

The algorithms converges in no more than \( \frac{p^{(1)} - p^*}{\eta} \) iteration.

Polynomial time convergence! For some common SVMs (like nu-SVM), the iteration can be a constant
The solution of WellSVM improves the safeness and scalability of previous solutions.
Experiment

RCV1: 47,236 features, 677,399 instances

The solution of WellSVM improves the safeness and scalability of previous solutions
Outline

- Improve the quality of optimization solution
  - WELL SVM [Li et al., JMLR13]

- Address the uncertainty of model selection
  - S4VM [Li and Zhou, TPAMI15]

- Overcome the variety of performance measures
  - UMVP [Li et al., AAAI16]
Observation

S3VMs

Large Margin Separator

i) **More than one** Large Margin Separators!!

ii) Current S3VMs randomly select one of them as the output.

iii) Large Margin Separators are usually **diverse**.

iv) **Incorrect selection** degenerates the performance!

We present S4VM (Safe S3VM) to address the uncertainty of model selection
S4VM: A simple algorithm

- Step 1: Generate a pool of large-margin separators (LMS).
- Step 2: Construct S4VM by optimizing the performance improvement under the worst-case.
S4VM Formulation

- Maximize accuracy

\[ \max_{y \in \{\pm 1\}^u} J(y, y^*, y^{svm}) = \text{gain}(y, y^*, y^{svm}) - \lambda \text{loss}(y, y^*, y^{svm}) \]

- \text{gain}(): gained accuracy against SVM (without using unlabeled data)
- \text{loss}(): lost accuracy against SVM (without using unlabeled data)
- \( \lambda \): measure the risk that user would like to undertake
- \( y^* \): ground-truth label assignment
- **Difficulty**: The ground-truth is unknown.

- Note that ground-truth is a LMS, we assume that \( y^* \in \{\hat{y}_t\}_{t=1}^T \)
- S4VM maximizes the worst-case accuracy

\[ \bar{y} = \arg \max_{y \in \{\pm 1\}^u} \min_{\hat{y} \in \{\hat{y}_t\}_{t=1}^T} \text{gain}(y, \hat{y}, y^{svm}) - \lambda \text{loss}(y, \hat{y}, y^{svm}) \]
Theoretical Analysis

Theorem 1: If \( y^* \in \{\hat{y}_t\}_{t=1}^T \) and \( \lambda \geq 1 \), the accuracy of \( \bar{y} \) is never worse than that of \( y^{svm} \).

Under the assumption employed in S3VMs, that is the ground-truth is realized by a large-margin separator, S4VM is provable safe.

Proposition 2: If \( y^* \in \{\hat{y}_t\}_{t=1}^T \) and \( \lambda = 1 \), the accuracy of \( \bar{y} \) achieves the maximal performance improvement over that of \( y^{svm} \) in the worst case.

Under the assumption employed in S3VMs, S4VM already achieves the largest performance improvement.
In terms of average performance, S4VM is highly competitive with TSVM.
Experiment

Significantly degenerated performance

<table>
<thead>
<tr>
<th>10 Lab</th>
<th>SVM</th>
<th>TSVM</th>
<th>S4VM_s</th>
</tr>
</thead>
<tbody>
<tr>
<td>aust</td>
<td>67.2±8.0/66.6±7.5</td>
<td>67.8±12.1/67.8±13.2</td>
<td>68.7±9.8/68.1±9.8</td>
</tr>
<tr>
<td>ausl</td>
<td>77.3±7.1/72.9±6.1</td>
<td><strong>81.1±6.1/77.0±8.2</strong></td>
<td>77.8±7.5/73.3±6.4</td>
</tr>
<tr>
<td>brea</td>
<td>94.4±3.5/95.7±2.8</td>
<td>94.5±0.4/94.7±0.1</td>
<td><strong>96.5±0.6/96.8±0.4</strong></td>
</tr>
<tr>
<td>clea</td>
<td>56.5±5.0/57.1±8.1</td>
<td>56.7±5.7/55.8±7.8</td>
<td>56.7±5.1/56.9±8.2</td>
</tr>
<tr>
<td>diab</td>
<td>65.7±5.4/65.6±5.5</td>
<td>65.7±5.4/65.3±6.2</td>
<td>66.4±5.2/65.5±6.2</td>
</tr>
<tr>
<td>germ</td>
<td>63.1±8.0/64.8±12.1</td>
<td>62.2±5.5/66.1±5.1</td>
<td>63.3±7.9/64.8±11.8</td>
</tr>
<tr>
<td>habe</td>
<td>64.6±7.0/65.9±8.1</td>
<td><strong>62.1±7.9/52.4±6.2</strong></td>
<td>65.0±6.5/65.9±7.9</td>
</tr>
<tr>
<td>hear</td>
<td>72.2±6.7/71.8±6.8</td>
<td>71.9±6.9/72.5±7.6</td>
<td>72.2±6.7/72.7±7.0</td>
</tr>
<tr>
<td>hous</td>
<td>89.0±3.0/88.0±2.8</td>
<td><strong>90.6±3.3/89.8±1.8</strong></td>
<td>89.6±3.1/88.5±2.3</td>
</tr>
<tr>
<td>houv</td>
<td>88.0±2.1/87.0±5.1</td>
<td>85.4±6.4/85.0±5.8</td>
<td>87.9±4.4/86.8±3.9</td>
</tr>
<tr>
<td>iono</td>
<td>72.6±6.8/74.7±9.1</td>
<td>72.1±9.4/77.4±8.6</td>
<td>73.7±6.7/75.2±10.0</td>
</tr>
<tr>
<td>isol</td>
<td>89.6±6.8/81.0±14.7</td>
<td>86.6±10.0/86.7±9.9</td>
<td><strong>91.4±9.0/85.2±17.2</strong></td>
</tr>
<tr>
<td>live</td>
<td>55.3±5.4/55.5±5.9</td>
<td>53.3±5.0/54.2±4.8</td>
<td>54.8±5.7/55.2±6.0</td>
</tr>
<tr>
<td>optd</td>
<td>92.2±4.4/85.9±3.8</td>
<td>87.1±8.7/87.2±8.7</td>
<td>93.5±5.6/87.9±11.3</td>
</tr>
<tr>
<td>vehi</td>
<td>74.0±7.9/74.3±8.1</td>
<td>76.2±9.2/78.7±8.7</td>
<td>74.7±5.5/76.2±9.8</td>
</tr>
<tr>
<td>wdbc</td>
<td>81.3±7.4/80.7±7.5</td>
<td><strong>84.3±6.6/84.4±6.3</strong></td>
<td>82.1±7.2/82.5±9.1</td>
</tr>
<tr>
<td>digit</td>
<td>76.2±7.1/57.2±12.1</td>
<td>80.7±3.9/84.2±4.4</td>
<td>76.0±7.1/63.6±12.6</td>
</tr>
<tr>
<td>USPS</td>
<td>78.5±2.9/80.0±0.1</td>
<td>72.2±3.7/71.6±4.9</td>
<td>78.7±2.7/80.1±0.2</td>
</tr>
<tr>
<td>COIL</td>
<td>56.5±5.4/56.9±4.2</td>
<td>54.9±6.2/57.2±3.9</td>
<td>56.8±5.4/57.0±4.2</td>
</tr>
<tr>
<td>BCI</td>
<td>52.6±2.4/51.3±2.0</td>
<td>50.9±2.8/51.2±2.1</td>
<td>51.8±2.3/51.3±2.2</td>
</tr>
<tr>
<td>g241c</td>
<td>54.5±4.2/52.3±4.4</td>
<td><strong>78.3±4.5/59.5±3.4</strong></td>
<td>54.6±4.5/52.8±4.4</td>
</tr>
<tr>
<td>g241n</td>
<td>56.4±5.2/52.5±5.3</td>
<td>53.3±7.3/52.8±5.2</td>
<td>56.3±5.1/52.7±5.1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Avg.Acc</th>
<th>SVM</th>
<th>TSVM</th>
<th>S4VM_s</th>
</tr>
</thead>
<tbody>
<tr>
<td>71.3/69.7</td>
<td>72.7/71.3</td>
<td>71.8/70.6</td>
<td>82.4/81.8</td>
</tr>
</tbody>
</table>

TSVM often degenerate the performance while S4VM does not significantly degenerate the performance.
Both S3VMs and S4VM assume that the ground-truth is realized by a large-margin separator.

**Assumption of S4VM**

<table>
<thead>
<tr>
<th>10 Lab</th>
<th>( \text{S3VM}_{best} ) (linear / rbf)</th>
<th>100 Lab</th>
<th>( \text{S3VM}_{best} ) (linear / rbf)</th>
</tr>
</thead>
<tbody>
<tr>
<td>aust</td>
<td>69.4 ± 10.0 / 73.1 ± 11.8</td>
<td>aust</td>
<td>83.7 ± 1.6 / 78.9 ± 2.8</td>
</tr>
<tr>
<td>ausi</td>
<td>82.2 ± 5.2 / 82.3 ± 4.0</td>
<td>ausi</td>
<td>81.5 ± 2.1 / 81.8 ± 1.9</td>
</tr>
<tr>
<td>brea</td>
<td>96.9 ± 0.3 / 96.8 ± 0.2</td>
<td>brea</td>
<td>95.2 ± 1.4 / 95.5 ± 1.0</td>
</tr>
<tr>
<td>clea</td>
<td>63.2 ± 4.7 / 61.9 ± 5.4</td>
<td>clea</td>
<td>74.1 ± 3.0 / 84.5 ± 1.8</td>
</tr>
<tr>
<td>diab</td>
<td>66.7 ± 5.2 / 65.8 ± 6.2</td>
<td>diab</td>
<td>74.4 ± 1.2 / 70.7 ± 2.3</td>
</tr>
<tr>
<td>germ</td>
<td>66.0 ± 4.5 / 65.9 ± 4.9</td>
<td>germ</td>
<td>66.2 ± 3.0 / 70.4 ± 1.7</td>
</tr>
<tr>
<td>habe</td>
<td>65.2 ± 6.1 / 65.7 ± 5.1</td>
<td>habe</td>
<td>71.6 ± 3.6 / 67.7 ± 2.6</td>
</tr>
<tr>
<td>hear</td>
<td>72.2 ± 6.8 / 74.4 ± 6.2</td>
<td>hear</td>
<td>82.6 ± 2.1 / 76.5 ± 3.6</td>
</tr>
<tr>
<td>houv</td>
<td>93.5 ± 4.0 / 90.1 ± 2.1</td>
<td>houv</td>
<td>94.5 ± 1.4 / 95.3 ± 1.7</td>
</tr>
<tr>
<td>iono</td>
<td>88.9 ± 5.6 / 87.0 ± 4.6</td>
<td>iono</td>
<td>91.2 ± 1.3 / 92.6 ± 1.8</td>
</tr>
<tr>
<td>isol</td>
<td>78.2 ± 5.8 / 80.8 ± 7.4</td>
<td>isol</td>
<td>83.6 ± 2.6 / 91.9 ± 2.0</td>
</tr>
<tr>
<td>live</td>
<td>90.0 ± 10.3 / 89.1 ± 12.1</td>
<td>live</td>
<td>98.9 ± 1.4 / 99.0 ± 1.5</td>
</tr>
<tr>
<td>optd</td>
<td>55.2 ± 6.0 / 55.6 ± 5.8</td>
<td>optd</td>
<td>63.2 ± 4.2 / 67.0 ± 3.4</td>
</tr>
<tr>
<td>vehi</td>
<td>76.1 ± 8.8 / 80.2 ± 10.8</td>
<td>vehi</td>
<td>93.7 ± 2.0 / 98.1 ± 0.8</td>
</tr>
<tr>
<td>wdbc</td>
<td>82.1 ± 7.1 / 84.2 ± 7.5</td>
<td>wdbc</td>
<td>94.6 ± 1.7 / 93.6 ± 1.7</td>
</tr>
<tr>
<td>digi</td>
<td>77.4 ± 4.6 / 82.5 ± 6.2</td>
<td>digi</td>
<td>93.7 ± 1.3 / 96.2 ± 1.1</td>
</tr>
<tr>
<td>USPS</td>
<td>81.6 ± 4.4 / 77.7 ± 3.1</td>
<td>USPS</td>
<td>89.1 ± 0.6 / 92.5 ± 2.0</td>
</tr>
<tr>
<td>COIL</td>
<td>65.6 ± 5.1 / 67.5 ± 5.9</td>
<td>COIL</td>
<td>83.2 ± 2.2 / 88.9 ± 2.4</td>
</tr>
<tr>
<td>BCI</td>
<td>53.9 ± 2.9 / 53.8 ± 1.8</td>
<td>BCI</td>
<td>70.7 ± 3.5 / 66.2 ± 2.7</td>
</tr>
<tr>
<td>g241c</td>
<td>60.6 ± 3.2 / 59.0 ± 3.0</td>
<td>g241c</td>
<td>78.5 ± 2.7 / 79.3 ± 2.7</td>
</tr>
<tr>
<td>g241n</td>
<td>60.3 ± 3.1 / 56.3 ± 2.9</td>
<td>g241n</td>
<td>74.8 ± 2.7 / 74.5 ± 4.3</td>
</tr>
<tr>
<td>Text</td>
<td>56.9 ± 2.4 / 55.3 ± 1.7</td>
<td>Text</td>
<td>71.3 ± 1.0 / 57.6 ± 3.8</td>
</tr>
</tbody>
</table>

Even the best LMS is far from the ground truth, but S4VMs still work well.

S4VM is quite robust.
Another good property is that S4VM considers the worst-case of multiple LMS, it is quite robust with the parameters.
Outline

• Improve the quality of optimization solution
  • WELL SVM [Li et al., JMLR 2013]

• Address the uncertainty of model selection
  • S4VM [Li and Zhou, TPAMI 2015]

• Overcome the variety of performance measures
  • UMVP [Li et al., AAAI 2016]
Variety of Performance Measures

- S4VM improves the safeness in terms of accuracy.
- real situations often require various performance measures.

For example
- In ranking applications
  - AUC
  - Top-k precision
- In text application
  - F1-Score
  - Precision-recall breakeven point
- In Information retrieval
  - Precision and recall
  - ...
Variety of Performance Measures

- The safeness in accuracy is not equal to the safeness in other performance measures.

- For example,

<table>
<thead>
<tr>
<th>Doc ID</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
</tr>
</thead>
<tbody>
<tr>
<td>( p )</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>rank(( h_1(x) ))</td>
<td>11</td>
<td>10</td>
<td>9</td>
<td>8</td>
<td>7</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>rank(( h_2(x) ))</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Hypothesis</th>
<th>MAP</th>
<th>Best Acc.</th>
</tr>
</thead>
<tbody>
<tr>
<td>( h_1(q) )</td>
<td>0.56</td>
<td>0.64</td>
</tr>
<tr>
<td>( h_2(q) )</td>
<td>0.51</td>
<td>0.73</td>
</tr>
</tbody>
</table>

develop “safe” SSL methods for various performance measures
UMVP

• Basic idea: Exploits multiple semi-supervised learners (SSLs) to derive a safe-aware SSL prediction

• Assume that we have trained multiple semi-supervised learners

\[ \{y^1, y^2, \ldots, y^b\} \]

• These learners could be obtained by
  • Different SSLs with different data assumption;
  • SSLs with different parameters;
  • A hybrid of the above two
UMVP Framework

- UMVP Framework: Maximize combined performance improvement over a baseline

\[
\max_{\hat{y} \in Y} \sum_{i=1}^{b} \alpha_i \left( \text{perf}(\hat{y}, y^i) - \text{perf}(\hat{y}^0, y^i) \right),
\]

Performance gain over baseline supervised model, when \( y^i \) is the ground-truth

- \( \alpha \) refers to the weight for learners

\[
\mathcal{M} = \{ \alpha \mid \sum_{i=1}^{b} \alpha_i = 1, \alpha_i \geq 0 \}.
\]

- \( \text{perf} \) refers to target performance measures, such as AUC, top-k precision, F1 score, etc.
UMVP Framework

- The weights of SSLs may not be available in practice

- Without further knowledge about the SSLs, we consider the worst-case

\[
\max_{\hat{y} \in \mathcal{Y}} \min_{\alpha \in \mathcal{M}} \sum_{i=1}^{b} \alpha_i \left( \text{perf}(\hat{y}, y^i) - \text{perf}(\hat{y}_0, y^i) \right).
\]

Challenging:

Non-convex non-continuous optimization

Performance gain over baseline supervised model, when \(y^i\) is the ground-truth
Cutting Plane Algorithm

- Convex Relaxation

\[
\min_{\alpha \in \mathcal{M}} \max_{\hat{y} \in \mathcal{Y}} \sum_{i=1}^{b} \alpha_i (\text{perf}(\hat{y}, y^i) - \text{perf}(\hat{y}_0, y^i)).
\]

- A tight convex relaxation

- Cutting-Plane Algorithm

\[
\min_{\alpha \in \mathcal{M}, \theta} \theta
\]

s.t. \[
\theta \geq \sum_{i=1}^{b} \alpha_i (\text{perf}(\hat{y}, y^i) - \text{perf}(\hat{y}_0, y^i)), \forall \hat{y} \in \mathcal{Y}.
\]
The key step in cutting plane optimization

\[
\arg \max_{\hat{y} \in Y} \sum_{i=1}^{b} \alpha_i (\text{perf}(\hat{y}, y^i) - \text{perf}(\hat{y}_0, y^i)).
\]

- Still non-convex and non-continues
- We show that when performance measure is Top-K precision, F1 and AUC, the above key step has a closed-form solution (can be solved efficient)
3.1.1 Top-$k$ Precision

**Proposition 1.** When the Top-$k$ precision is used in Eq.(5), any $\hat{y}$ that ranks the unlabeled instances identically as $s = \sum_{i=1}^{b} \alpha_i y^i$ is optimal (ties are broken arbitrarily).

3.1.2 $F_\beta$ Score

**Proposition 2.** Assume that $\hat{y}'1 = c$, a constant. Let $s = \sum_{i=1}^{b} \frac{\alpha_i(1+\beta^2)}{c+\beta^2 P_{y^i}} y^i$, where $P_{y^i} = y^i'1$ is the number of positive labels in $y^i$. When the $F_\beta$ score is used in Eq.(5), the optimal $\hat{y}^* = [\hat{y}^*_j]$ is given by

$$\hat{y}^*_j = \begin{cases} 1 & \pi_j^s > u - c \\ 0 & \text{otherwise} \end{cases}.$$  

3.1.3 AUC

**Proposition 3.** When the AUC is used in Eq.(5), any $\hat{y}$ that ranks the unlabeled instances identically as $s = \sum_{i=1}^{b} \frac{\alpha_i}{P_{y^i} N_{y^i}} y^i$ is optimal.
Experiment

<table>
<thead>
<tr>
<th>Data</th>
<th># sample</th>
<th># feature</th>
<th># pos/# neg</th>
<th>Data</th>
<th># sample</th>
<th># feature</th>
<th># pos/# neg</th>
</tr>
</thead>
<tbody>
<tr>
<td>COIL2</td>
<td>1,500</td>
<td>241</td>
<td>1.0</td>
<td>mnist7 vs 9</td>
<td>14,251</td>
<td>600</td>
<td>1.05</td>
</tr>
<tr>
<td>digit1</td>
<td>1,500</td>
<td>241</td>
<td>0.96</td>
<td>mnist1 vs 7</td>
<td>15,170</td>
<td>652</td>
<td>1.08</td>
</tr>
<tr>
<td>ethn</td>
<td>2,630</td>
<td>30</td>
<td>0.99</td>
<td>adult-a</td>
<td>32,561</td>
<td>123</td>
<td>0.32</td>
</tr>
<tr>
<td>mnist4 vs 9</td>
<td>13,782</td>
<td>629</td>
<td>0.98</td>
<td>w8a</td>
<td>49,749</td>
<td>300</td>
<td>0.03</td>
</tr>
<tr>
<td>mnist3 vs 8</td>
<td>13,966</td>
<td>631</td>
<td>1.05</td>
<td>real-sim</td>
<td>72,309</td>
<td>20,958</td>
<td>0.44</td>
</tr>
</tbody>
</table>

cover a wide range of properties

- Data size from 1,500 to more than 70,000
- Dimensionality from 30 to more than 20,000
- The proportion of classes (i.e., ratio of the number of positive samples to that of negative samples) ranges from 0.03 to around 1
Three Aspects of Safeness

By comparing with baseline supervised model, we use three aspects to describe the safeness of SSL methods

- Average performance improvement
  - the ability of SSL methods in performance improvement

- Win/Tie/Loss
  - the degree of performance degradation of SSL methods

- Sign-test
  - the dependence between the performance of SSL methods and data sets.
<table>
<thead>
<tr>
<th></th>
<th>Pre@k</th>
<th>F1</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Average performance</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>improvement</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Self-SVM$^{perf}$</td>
<td>-2.5</td>
<td>3.7</td>
<td>-0.9</td>
</tr>
<tr>
<td>S4VM</td>
<td>1.0</td>
<td>5.3</td>
<td>-0.5</td>
</tr>
<tr>
<td>UMVP$^{-}$</td>
<td>1.7</td>
<td>6.0</td>
<td>0.3</td>
</tr>
<tr>
<td>UMVP</td>
<td>1.8</td>
<td>5.9</td>
<td>0.8</td>
</tr>
<tr>
<td><strong>Win/Tie/Loss</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Self-SVM$^{perf}$</td>
<td>2/2/6</td>
<td>8/1/1</td>
<td>2/2/6</td>
</tr>
<tr>
<td>S4VM</td>
<td>4/5/1</td>
<td>8/1/1</td>
<td>6/1/3</td>
</tr>
<tr>
<td>UMVP$^{-}$</td>
<td>6/3/1</td>
<td>8/1/1</td>
<td>7/0/3</td>
</tr>
<tr>
<td>UMVP</td>
<td>6/4/0</td>
<td>8/1/1</td>
<td>8/2/0</td>
</tr>
<tr>
<td><strong>Sign test (H, p-value)</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Self-SVM$^{perf}$</td>
<td>(0, 0.29)</td>
<td>(1, 0.04)</td>
<td>(0, 0.29)</td>
</tr>
<tr>
<td>S4VM</td>
<td>(0, 0.38)</td>
<td>(1, 0.04)</td>
<td>(0, 0.51)</td>
</tr>
<tr>
<td>UMVP$^{-}$</td>
<td>(0, 0.13)</td>
<td>(1, 0.04)</td>
<td>(0, 0.34)</td>
</tr>
<tr>
<td>UMVP</td>
<td>(1, 0.03)</td>
<td>(1, 0.04)</td>
<td>(1, 0.01)</td>
</tr>
</tbody>
</table>

On average performance improvement, UMVP achieves performance improvement on all the three performance measures.
### Experiment

<table>
<thead>
<tr>
<th>Average performance improvement</th>
<th>Pre@k</th>
<th>F1</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Self-SVM&lt;sub&gt;perf&lt;/sub&gt;</td>
<td>-2.5</td>
<td>3.7</td>
<td>-0.9</td>
</tr>
<tr>
<td>S4VM</td>
<td>1.0</td>
<td>5.3</td>
<td>-0.5</td>
</tr>
<tr>
<td>UMVP&lt;sup&gt;−&lt;/sup&gt;</td>
<td>1.7</td>
<td>6.0</td>
<td>0.3</td>
</tr>
<tr>
<td>UMVP</td>
<td>1.8</td>
<td>5.9</td>
<td>0.8</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Win/Tie/Loss</th>
<th>Pre@k</th>
<th>F1</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Self-SVM&lt;sub&gt;perf&lt;/sub&gt;</td>
<td>2/2/6</td>
<td>8/1/1</td>
<td>2/2/6</td>
</tr>
<tr>
<td>S4VM</td>
<td>4/5/1</td>
<td>8/1/1</td>
<td>6/1/3</td>
</tr>
<tr>
<td>UMVP&lt;sup&gt;−&lt;/sup&gt;</td>
<td>6/3/1</td>
<td>8/1/1</td>
<td>7/0/3</td>
</tr>
<tr>
<td><strong>UMVP</strong></td>
<td>6/4/0</td>
<td>8/1/1</td>
<td>8/2/0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sign test (H, p-value)</th>
<th>Pre@k</th>
<th>F1</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Self-SVM&lt;sub&gt;perf&lt;/sub&gt;</td>
<td>(0, 0.29)</td>
<td>(1, 0.04)</td>
<td>(0, 0.29)</td>
</tr>
<tr>
<td>S4VM</td>
<td>(0, 0.38)</td>
<td>(1, 0.04)</td>
<td>(0, 0.51)</td>
</tr>
<tr>
<td>UMVP&lt;sup&gt;−&lt;/sup&gt;</td>
<td>(0, 0.13)</td>
<td>(1, 0.04)</td>
<td>(0, 0.34)</td>
</tr>
<tr>
<td>UMVP</td>
<td>(1, 0.03)</td>
<td>(1, 0.04)</td>
<td>(1, 0.01)</td>
</tr>
</tbody>
</table>

In Win/Tie/Loss, each of the comparison methods leads to significant drops in performance in at least 5 cases, while the UMVP method only has one.

In addition, the UMVP method achieves significant improvement in 22 cases, which is the most among all the methods.
## Experiment

<table>
<thead>
<tr>
<th>Average performance improvement</th>
<th>Pre@k</th>
<th>F1</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Self-SVM$^{perf}$</td>
<td>-2.5</td>
<td>3.7</td>
<td>-0.9</td>
</tr>
<tr>
<td>S4VM</td>
<td>1.0</td>
<td>5.3</td>
<td>-0.5</td>
</tr>
<tr>
<td>UMVP$^{-}$</td>
<td>1.7</td>
<td>6.0</td>
<td>0.3</td>
</tr>
<tr>
<td>UMVP</td>
<td>1.8</td>
<td>5.9</td>
<td>0.8</td>
</tr>
<tr>
<td>Win/Tie/Loss</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Self-SVM$^{perf}$</td>
<td>2/2/6</td>
<td>8/1/1</td>
<td>2/2/6</td>
</tr>
<tr>
<td>S4VM</td>
<td>4/5/1</td>
<td>8/1/1</td>
<td>6/1/3</td>
</tr>
<tr>
<td>UMVP$^{-}$</td>
<td>6/3/1</td>
<td>8/1/1</td>
<td>7/0/3</td>
</tr>
<tr>
<td>UMVP</td>
<td>6/4/0</td>
<td>8/1/1</td>
<td>8/2/0</td>
</tr>
<tr>
<td>Sign test (H, p-value)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Self-SVM$^{perf}$</td>
<td>(0, 0.29)</td>
<td>(1, 0.04)</td>
<td>(0, 0.29)</td>
</tr>
<tr>
<td>S4VM</td>
<td>(0, 0.38)</td>
<td>(1, 0.04)</td>
<td>(0, 0.51)</td>
</tr>
<tr>
<td>UMVP$^{-}$</td>
<td>(0, 0.13)</td>
<td>(1, 0.04)</td>
<td>(0, 0.34)</td>
</tr>
<tr>
<td>UMVP</td>
<td>(1, 0.03)</td>
<td>(1, 0.04)</td>
<td>(1, 0.01)</td>
</tr>
</tbody>
</table>

In the statistical significance test (using the Wilcoxon sign test at 95% significance level) of 10 data sets, the UMVP method is superior to baseline supervised model on all the three performance measures, while the other comparison methods do not obtain such a significance.
## Training Time

<table>
<thead>
<tr>
<th></th>
<th>$\text{SVM}^{perf}$</th>
<th>$\text{Self-SVM}^{perf}$</th>
<th>$\text{S4VM}$</th>
<th>UMVP</th>
</tr>
</thead>
<tbody>
<tr>
<td>adult-a</td>
<td>0.844</td>
<td>145.516</td>
<td>22.403</td>
<td>34.811 (32.936 + 1.875)</td>
</tr>
<tr>
<td>mnist3vs8</td>
<td>3.622</td>
<td>621.665</td>
<td>148.980</td>
<td>87.891 (87.435 + 0.456)</td>
</tr>
<tr>
<td>mnist7vs9</td>
<td>3.093</td>
<td>638.300</td>
<td>116.440</td>
<td>72.622 (72.155 + 0.467)</td>
</tr>
<tr>
<td>mnist1vs7</td>
<td>2.791</td>
<td>465.190</td>
<td>101.235</td>
<td>57.697 (57.220 + 0.477)</td>
</tr>
<tr>
<td>mnist4vs9</td>
<td>3.411</td>
<td>597.095</td>
<td>121.038</td>
<td>87.179 (86.765 + 0.414)</td>
</tr>
<tr>
<td>real-sim</td>
<td>7.975</td>
<td>1073.755</td>
<td>93.880</td>
<td>129.196 (119.552 + 9.644)</td>
</tr>
<tr>
<td>w8a</td>
<td>1.486</td>
<td>888.995</td>
<td>35.172</td>
<td>38.985 (35.091 + 3.894)</td>
</tr>
<tr>
<td>ethn</td>
<td>0.247</td>
<td>9.737</td>
<td>2.074</td>
<td>3.521 (3.458 + 0.063)</td>
</tr>
<tr>
<td>COIL2</td>
<td>0.698</td>
<td>16.593</td>
<td>20.114</td>
<td>11.506 (11.466 + 0.04)</td>
</tr>
<tr>
<td>digit1</td>
<td>0.699</td>
<td>22.700</td>
<td>20.342</td>
<td>11.472 (11.430 + 0.042)</td>
</tr>
</tbody>
</table>

Although most of the time of UMVP is spent on generating the SSLs, the optimization part of UMVP is fast (The number of iterations required in Algorithm 1 is usually fewer than 100). Overall, the training time of UMVP is comparable with S4VM but more efficient than Self-SVM$_{perf}$. 
Summary

Study safe SSL

- WELL SVM
  - Tries to improve the quality of optimization solution
  - Empirical studies show that the solution of WELL SVM improves the safeness and scalability of previous solutions
  - [http://lamda.nju.edu.cn/code_WELL SVM.ashx](http://lamda.nju.edu.cn/code_WELL SVM.ashx)

- S4 VM
  - Tries to address the uncertainty of model selection
  - Empirical studies show that S4VM significantly improve the safeness in terms of accuracy
  - [http://lamda.nju.edu.cn/code_S4VM.ashx](http://lamda.nju.edu.cn/code_S4VM.ashx)

- UMVP
  - Tries to overcome the variety of performance measures
  - Empirical studies show that UMVP improve the safeness for various performance measures

Thanks!