Large Scale Semi-supervised Linear SVMs

Vikas Sindhwani and Sathiya Keerthi

University of Chicago

Yahoo! Research

SIGIR 2006
Semi-supervised Learning (SSL)

Motivation

- Categorize x-billion documents into commercial/non-commercial.
- Traditional machine learning algorithms require labels.
- Labels are expensive/impossible to get.
- But tons of unlabeled data!

Setting

Linear SVMs ($S^3$VM) for large-scale problems – large number of examples and features – highly sparse – few labels and lots of unlabeled data.
Semi-supervised Learning (SSL)

Motivation

- Categorize x-billion documents into commercial/non-commercial.
- Traditional machine learning algorithms require labels.
- Labels are expensive/impossible to get.
- But tons of unlabeled data!

Setting

Linear SVMs ($S^3$VM) for large-scale problems – large number of examples and features – highly sparse – few labels and lots of unlabeled data.
Outline

1 Fast (fully supervised) Linear SVMs

2 The cluster assumption for SSL

3 Semi-supervised SVMs
   - An objective function to implement cluster assumption
   - A Scalable Label-switching Algorithm
   - The Problem of Non-convexity
   - A Deterministic Annealing (DA) approach

4 Empirical Studies

5 Extensions
Outline

1. Fast (fully supervised) Linear SVMs
2. The cluster assumption for SSL
3. Semi-supervised SVMs
   - An objective function to implement cluster assumption
   - A Scalable Label-switching Algorithm
   - The Problem of Non-convexity
   - A Deterministic Annealing (DA) approach
4. Empirical Studies
5. Extensions
Outline

1. Fast (fully supervised) Linear SVMs
2. The cluster assumption for SSL
3. Semi-supervised SVMs
   - An objective function to implement cluster assumption
   - A Scalable Label-switching Algorithm
   - The Problem of Non-convexity
   - A Deterministic Annealing (DA) approach
4. Empirical Studies
5. Extensions
Outline

1. Fast (fully supervised) Linear SVMs
2. The cluster assumption for SSL
3. Semi-supervised SVMs
   - An objective function to implement cluster assumption
   - A Scalable Label-switching Algorithm
   - The Problem of Non-convexity
   - A Deterministic Annealing (DA) approach
4. Empirical Studies
5. Extensions
Outline

1. Fast (fully supervised) Linear SVMs
2. The cluster assumption for SSL
3. Semi-supervised SVMs
   - An objective function to implement cluster assumption
   - A Scalable Label-switching Algorithm
   - The Problem of Non-convexity
   - A Deterministic Annealing (DA) approach
4. Empirical Studies
5. Extensions
Outline

1. Fast (fully supervised) Linear SVMs
2. The cluster assumption for SSL
3. Semi-supervised SVMs
   - An objective function to implement cluster assumption
   - A Scalable Label-switching Algorithm
   - The Problem of Non-convexity
4. Empirical Studies
5. Extensions
Outline

1. Fast (fully supervised) Linear SVMs
2. The cluster assumption for SSL
3. Semi-supervised SVMs
   - An objective function to implement cluster assumption
   - A Scalable Label-switching Algorithm
   - The Problem of Non-convexity
   - A Deterministic Annealing (DA) approach

4. Empirical Studies
5. Extensions
Outline

1. Fast (fully supervised) Linear SVMs
2. The cluster assumption for SSL
3. Semi-supervised SVMs
   - An objective function to implement cluster assumption
   - A Scalable Label-switching Algorithm
   - The Problem of Non-convexity
   - A Deterministic Annealing (DA) approach
4. Empirical Studies
5. Extensions
Outline

1. Fast (fully supervised) Linear SVMs
2. The cluster assumption for SSL
3. Semi-supervised SVMs
   - An objective function to implement cluster assumption
   - A Scalable Label-switching Algorithm
   - The Problem of Non-convexity
   - A Deterministic Annealing (DA) approach
4. Empirical Studies
5. Extensions
<table>
<thead>
<tr>
<th>Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
</tr>
<tr>
<td>Fast (fully supervised) Linear SVMs</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>The cluster assumption for SSL</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>Semi-supervised SVMs</td>
</tr>
<tr>
<td>• An objective function to implement cluster assumption</td>
</tr>
<tr>
<td>• A Scalable Label-switching Algorithm</td>
</tr>
<tr>
<td>• The Problem of Non-convexity</td>
</tr>
<tr>
<td>• A Deterministic Annealing (DA) approach</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>Empirical Studies</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>Extensions</td>
</tr>
</tbody>
</table>
Fast Linear \((l_2)\)-SVMs [Keerthi & Decoste, 2005]

Given \(\{x_i \in \mathbb{R}^d, y_i = \pm 1\}_{i=1}^l\), data matrix \(X\) \((l \times d)\) is sparse.

**Optimization**

\[
\min_{w \in \mathbb{R}^d} \frac{\lambda}{2} \|w\|^2 + \sum_{i=1}^l l_2 \left( y_i, w^T x_i \right)
\]

- continuously differentiable, whereas standard \(l_1\) loss is not differentiable.
- Primal, unconstrained, direct \(w\) optimization, whereas LIBSVM/SVM-light are dual methods.
- Only \(X \times \text{vec}\) operations, whereas dual methods deal with dense gram matrix.

\(l_2\) (Squared hinge Loss)
Fast Linear ($l_2$)-SVMs  [Keerthi & Decoste, 2005]

Algorithm

$$\min_{w \in \mathbb{R}^d} J(w) = \frac{\lambda}{2} \|w\|^2 + \sum_{i \in A(w)} c_i \left(1 - y_i(w^T x_i)\right)^2$$

where $A(w) = \{i : y_i(w^T x_i) < 1\}$

- Initialize $w_0$
- Iterate: $k=0,1,2,...$
  - Regularized Least Squares:
    $$\tilde{w} = \min_w \frac{\lambda}{2} \|w\|^2 + \sum_{i \in A(w_k)} c_i \left(1 - y_i(w^T x_i)\right)^2$$
  - Set search direction $d = \tilde{w} - w_k$
  - Line Search: Solve $\delta^* = \min_{\delta} J(w_k + \delta d)$
  - Set new iterate: $w_{k+1} = w_k + \delta^*(\tilde{w} - w_k)$
### Fast Linear ($l_2$)-SVMs [Keerthi & Decoste, 2005]

**Specialized Conjugate gradient (CGLS) to solve RLS**

To get $\tilde{w}$, Minimize:

$$\frac{1}{2} w^T [X^T C X + \lambda I] w - [X^T C Y] w$$

where $X$: data matrix (rows are examples), $C$: diagonal cost matrix, $Y$: label vector – only over $A(w_k)$

- $|A(w_k)|$ may be much smaller than $l$.
- Use $w_k$ as the initial seed. Seeding very effective.
- Only operations involving $X$ are matrix-vector products of the form $Xp$ and $X^Tz$ – can be done fast.

### Typical Behaviour: Reuters CCAT

Finite convergence guaranteed.

804414 examples, 47256 features: #CGLS iterations (10, 15, 8, 2 ; 28, 19) $\rightarrow$ 7 iterations, Total 80 seconds [3GHz, 2GB]
Outline

1. Fast (fully supervised) Linear SVMs
2. The cluster assumption for SSL
3. Semi-supervised SVMs
   - An objective function to implement cluster assumption
   - A Scalable Label-switching Algorithm
   - The Problem of Non-convexity
   - A Deterministic Annealing (DA) approach
4. Empirical Studies
5. Extensions
Cluster Assumption

Assumptions

Two points in a cluster have same labels.

Design Principle: Drive the classification hyperplane away from the data – while respecting labels. Decisions should not change within a cluster.
Cluster Assumption

Assumptions

Two points in a cluster have the same labels.

Design Principle: Drive the classification hyperplane away from the data – while respecting labels. Decisions should not change within a cluster.
Cluster Assumption

Assumptions

Two points in a cluster have same labels.

Design Principle: Drive the classification hyperplane away from the data – while respecting labels. Decisions should not change within a cluster.
Cluster Assumption

Assumptions

Two points in a cluster have the same labels.

**Design Principle:** Drive the classification hyperplane away from the data – while respecting labels. Decisions should not change within a cluster.
Outline

1. Fast (fully supervised) Linear SVMs
2. The cluster assumption for SSL
3. Semi-supervised SVMs
   - An objective function to implement cluster assumption
   - A Scalable Label-switching Algorithm
   - The Problem of Non-convexity
   - A Deterministic Annealing (DA) approach
4. Empirical Studies
5. Extensions
Vapnik’s idea

An objective function to implement cluster assumption

Given $l$ labeled examples $\{x_i, y_i\}_{i=1}^{l}$, $u$ unlabeled examples $x'_j$.

Train an SVM while optimizing unknown labels

Solve for weights $w$ and unknown labels $y' \in \{-1, +1\}^u$,

$$
\text{min}_{w, y'} \frac{\lambda}{2} \|w\|^2 + \frac{1}{l} \sum_{i=1}^{l} l_2 \left(y_i, w^T x_i\right) + \frac{\lambda'}{u} \sum_{j=1}^{u} l_2 \left(y'_j, w^T x'_j\right)
$$

subject to: $\frac{1}{u} \sum_{j=1}^{u} \max(0, y'_j) = r$ (positive class ratio)
An objective function to implement cluster assumption

Equivalent Problem

Optimization Problem

$$\min_{w, y'} J(w, y') = \frac{\lambda}{2} \|w\|^2 + \frac{1}{l} \sum_{i=1}^{l} l_2 (y_i, o_i) + \frac{\lambda'}{u} \sum_{j=1}^{u} l_2 (y'_j, o'_j)$$

$$\min_{w} J(w) = \frac{\lambda}{2} \|w\|^2 + \frac{1}{l} \sum_{i=1}^{l} l_2 (y_i, o_i) + \frac{\lambda'}{u} \sum_{j=1}^{u} \min_j \left[ l_2 (+, o'_j), l_2 (-, o'_j) \right]$$

This represents the effective loss $l'_2(o'_j)$.
Effective Loss Function Over Unlabeled Examples

An objective function to implement cluster assumption

- Non-convex
- Penalty if decision surface gets too close to unlabeled examples.
Fast TSVMs

**SVM-Light Implementation**

- Train an SVM on labeled data.
- Initialize $y'$ by labeling unlabeled data (fraction $r$ positive).
- Iterate:
  - Optimize $w$ keeping $y'$ fixed
    - Train SVM using **SVM-Light** with $y'$ as labels of unlabeled data.
  - Optimize $y'$ keeping $w$ fixed
    - Switch a pair of labels so that objective function strictly decreases.
Fast (fully supervised) Linear SVMs

The cluster assumption for SSL

Semi-supervised SVMs

Empirical Studies

Extensions

A Scalable Label-switching Algorithm

Fast T SVMs

Our Implementation

- Train an SVM on labeled data.
- Initialize \( y' \) by labeling unlabeled data (fraction \( r \) positive).
- Iterate:
  - Optimize \( w \) keeping \( y' \) fixed
    - Train SVM using Fast \( l_2 \)-SVM with \( y' \) as labels of unlabeled data. Seed previous \( w \).
  - Optimize \( y' \) keeping \( w \) fixed
    - Switch a pair of labels so that objective function strictly decreases.
Our Implementation

- Train an SVM on labeled data.
- Initialize $y'$ by labeling unlabeled data (fraction $r$ positive).
- Iterate:
  - Optimize $w$ keeping $y'$ fixed
    - Train SVM using \textit{Fast $l_2$-SVM} with $y'$ as labels of unlabeled data. \textit{Seed previous} $w$.
  - Optimize $y'$ keeping $w$ fixed
    - Switch \textit{multiple pairs} of labels so that objective function \textit{strictly} decreases.
Our Implementation

- Train an SVM on labeled data.
- Initialize $y'$ by labeling unlabeled data (fraction $r$ positive).
- Iterate:
  - Optimize $w$ keeping $y'$ fixed
    - Train SVM using Fast $l_2$-SVM with $y'$ as labels of unlabeled data. Seed previous $w$.
  - Optimize $y'$ keeping $w$ fixed
    - Switch multiple pairs of labels so that objective function strictly decreases.

Question: Termination guaranteed – but how many switches and how efficient will it be to retrain so many times?
A Scalable Label-switching Algorithm

Label Switching

- Sort (currently) + examples by margin error. ● Sort (currently) – examples by margin error. ● Switch S pairs or until sum of margin errors falls below 2.
Fast (fully supervised) Linear SVMs

The cluster assumption for SSL

Semi-supervised SVMs

Empirical Studies

Extensions

A Scalable Label-switching Algorithm

Label Switching

- Sort (currently) \(+\) examples by margin error.
- Sort (currently) \(-\) examples by margin error.
- Switch $S$ pairs or until sum of margin errors falls below 2.
A Scalable Label-switching Algorithm

Label Switching

- Sort (currently) $+$ examples by margin error.
- Sort (currently) $-$ examples by margin error.
- Switch $S$ pairs or until sum of margin errors falls below 2.
The Problem of Non-convexity

Non-convexity can hurt empirical performance
The Problem of Non-convexity

Non-convexity can hurt empirical performance
The Problem of Non-convexity

Non-convexity can hurt empirical performance
The Problem of Non-convexity

Non-convexity can hurt empirical performance
The Problem of Non-convexity

Non-convexity can hurt empirical performance

- Objective Function
- \( \theta \)
The Problem of Non-convexity

Non-convexity can hurt empirical performance
The Problem of Non-convexity

Non-convexity can hurt empirical performance
Non-convexity can hurt empirical performance

Error rates on COIL6: SVM 21.9, TSVM 21.2, ▽ TSVM 21.6
Handling Local Minima

- Start with an easy (unimodal) objective function and gradually increase non-convexity.
- Work with a family of objective functions; parameterically track minimizers.
- $J_{\chi}$ is insensitive to \textit{outside} unlabeled data.
Handling Local Minima

- Start with an easy (unimodal) objective function and gradually increase non-convexity.
- Work with a family of objective functions; parameterically track minimizers.
- $J_{\lambda'}(w, y') = \frac{\lambda}{2} \|w\|^2 + \frac{1}{l} \sum_{i=1}^{l} l_2(y_i, o_i) + \frac{\lambda'}{u} \sum_{j=1}^{u} l_2(y'_j, o'_j)$

Effective loss

Increasing $\lambda'$

$J_{\lambda'}$ is insensitive to outside unlabeled data.
The Problem of Non-convexity

Handling Local Minima

- Start with an easy (unimodal) objective function and gradually increase non-convexity.
- Work with a family of objective functions; parameterically track minimizers.
- $J_{\lambda'}$ is insensitive to outside unlabeled data.

$$J_{\lambda'}(w, y') = \frac{\lambda'}{2} \| w \|^2 + \frac{1}{T} \sum_{i=1}^{T} l_2(y_i, o_i) + \frac{\lambda'}{u} \sum_{j=1}^{u} l_2(y'_j, o'_j)$$

Effective loss

Increasing $\lambda'$

Effective loss
Deterministic Annealing: Intuition

**Question**

For the decision boundary to locally evolve in a desirable manner, what should the loss function look like?

![Graph showing the decision boundary and loss function.](image)

**Key Idea**

Deform the loss function (objective) as the optimization proceeds; use outside unlabeled data.
The Problem of Non-convexity

**Deterministic Annealing: Intuition**

**Question**

For the decision boundary to locally evolve in a desirable manner, what should the loss function look like?

![Graphs showing the decision boundary and loss function](Image)

**Key Idea**

Deform the loss function (objective) as the optimization proceeds; use *outside* unlabeled data.
Deterministic Annealing: Intuition

**Question**

For the decision boundary to locally evolve in a desirable manner, what should the loss function look like?

![Graphs showing the decision boundary and loss function evolution.](image)

**Key Idea**

Deform the loss function (objective) as the optimization proceeds; use *outside* unlabeled data.
A Deterministic Annealing (DA) approach

Deterministic Annealing for Semi-supervised SVMs

Another Equivalent Continuous Optimization Problem

“Relax” $y'$ to $p = (p_1 \ldots p_u)$ where $p_j$ is like the prob that $y'_j = 1$.

$$J(w, p) = E_p J(w, y') = \frac{\lambda}{2} \|w\|^2 + \frac{1}{l} \sum_{i=1}^{l} l_2(y_i, o_i)$$

$$+ \frac{\chi'}{u} \sum_{j=1}^{u} \left[ p_j l_2(+, o'_j) + (1 - p_j) l_2(-, o'_j) \right]$$

Family of Objective Functions: Avg Cost - T Entropy

$$J_T(w, p) = E_p J(w, y') - \underbrace{T \, H(p)}_{\text{$\text{Avg Cost}$}}$$

$$- \frac{T}{u} \sum_{j=1}^{u} [p_j \log p_j + (1 - p_j) \log (1 - p_j)]$$
A Deterministic Annealing (DA) approach

Deterministic Annealing: Some Quick Comments

**Smoothing Interpretation**

At high $T$, spurious & shallow local min are smoothed away.

**Deterministic Variant of Simulated Annealing (SA)**

SA is a stochastic search technique based on setting up a Markov process whose steady state distribution minimizes $E_p J - TH(p)$. Probabilistic guarantee for global optimum as $T \to 0$ very slowly.

**Proven Heuristic**

No guarantees, but has a strong record of empirical success.
A Deterministic Annealing (DA) approach

Deterministic Annealing: Some Quick Comments

Smoothing Interpretation
At high $T$, spurious & shallow local min are smoothed away.

Deterministic Variant of Simulated Annealing (SA)
SA is a stochastic search technique based on setting up a Markov process whose steady state distribution minimizes $E_p J - TH(p)$. Probabilistic guarantee for global optimum as $T \to 0$ very slowly.

Proven Heuristic
No guarantees, but has a strong record of empirical success.
A Deterministic Annealing (DA) approach

Deterministic Annealing: Some Quick Comments

### Smoothing Interpretation

At high $T$, spurious & shallow local min are smoothed away.

### Deterministic Variant of Simulated Annealing (SA)

SA is a stochastic search technique based on setting up a Markov process whose steady state distribution minimizes $E_p J - TH(p)$. Probabilistic guarantee for global optimum as $T \to 0$ very slowly.

### Proven Heuristic

No guarantees, but has a strong record of empirical success.
A Deterministic Annealing (DA) approach

Deterministic Annealing for Semi-supervised SVMs

**Full Optimization problem at T**

\[
\min_{w,p} J_T(w, p) = \frac{\lambda}{2} \|w\|^2 + \frac{1}{T} \sum_{i=1}^l l_2 (y_i, o_i) + \frac{\lambda'}{u} \sum_{j=1}^u \left[ p_j l_2 (+, o_j') + (1 - p_j) l_2 (-, o_j') \right] + \frac{T}{u} \sum_{j=1}^u [p_j \log p_j + (1 - p_j) \log p_j] \quad \text{s.t.} \quad (1/u) \sum_{j=1}^u p_j = r
\]

**Details**

- **Deformation:** $T$ controls non-convexity of $J_T(w, p)$. At $T = 0$, reduces to the original non-convex objective function $J(w, p)$.
- **Optimization at** $T$ $(w_T^*, p_T^*) = \arg\min_{w,p} J_T(w, p)$
- **Annealing:** Return: $w^* = \lim_{T \to 0} w_T^*$
- **Balance constraint:** $\frac{1}{u} \sum_{j=1}^u p_j = r$
A Deterministic Annealing (DA) approach

Alternating Convex Optimization

At any $T$, optimize $w$ keeping $p$ fixed

- Use Fast $l_2$-SVMs – two copies of unlabeled data but due to linearity, can reformulate CGLS to work on one.

At any $T$, optimize $p$ keeping $w$ fixed

- $p^*_j = \frac{1}{g_j - \nu} \quad g_j = \lambda' \left[ l_2(+, o'_j) - l_2(-, o'_j) \right]

- Obtain $\nu$ by solving $\frac{1}{u} \sum_{j=1}^{u} \frac{1}{\frac{g_j - \nu}{1 + e^{-\frac{T}{\nu}}}} = r$

Stopping Conditions

- At any $T$, alternate until $KL(p_{\text{new}} | p_{\text{old}}) < \epsilon$. Obtain $p^*_T$.
- Reduce $T$, Seed old $p^*_T$, until $H(p^*_T) < \epsilon$. 
A Deterministic Annealing (DA) approach

**DA Effective Loss wrt $T$**

**Effective DA l2 Loss**

![Graph showing the Effective DA l2 Loss](image-url)
Outline

1. Fast (fully supervised) Linear SVMs
2. The cluster assumption for SSL
3. Semi-supervised SVMs
   - An objective function to implement cluster assumption
   - A Scalable Label-switching Algorithm
   - The Problem of Non-convexity
   - A Deterministic Annealing (DA) approach
4. Empirical Studies
5. Extensions
## Experiments

<table>
<thead>
<tr>
<th>Dataset</th>
<th>#features</th>
<th>sparsity</th>
<th>#train</th>
<th>#test</th>
<th>r</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>aut-avn</td>
<td>20707</td>
<td>51.32</td>
<td>35588</td>
<td>35587</td>
<td>0.65</td>
<td>Usenet</td>
</tr>
<tr>
<td>real-sim</td>
<td>20958</td>
<td>51.32</td>
<td>36155</td>
<td>36154</td>
<td>0.31</td>
<td>Usenet</td>
</tr>
<tr>
<td>ccat</td>
<td>47236</td>
<td>75.93</td>
<td>17332</td>
<td>5787</td>
<td>0.46</td>
<td>Reuters</td>
</tr>
<tr>
<td>gcat</td>
<td>47236</td>
<td>75.93</td>
<td>17332</td>
<td>5787</td>
<td>0.30</td>
<td>Reuters</td>
</tr>
<tr>
<td>33-36</td>
<td>59072</td>
<td>26.56</td>
<td>41346</td>
<td>41346</td>
<td>0.49</td>
<td>Yahoo!</td>
</tr>
<tr>
<td>pcmac</td>
<td>7511</td>
<td>54.58</td>
<td>1460</td>
<td>486</td>
<td>0.51</td>
<td>20NG</td>
</tr>
<tr>
<td>rcv1</td>
<td>47236</td>
<td>76.73</td>
<td>804414</td>
<td>-</td>
<td>0.18</td>
<td>Reuters</td>
</tr>
</tbody>
</table>
Quality of Optimization

<table>
<thead>
<tr>
<th>DA</th>
<th>TSVM (1)</th>
<th>TSVM (max)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Observations**

- DA often gives significantly better minimizers [aut-avn, ccat, pcmac].
- Multiple switching TSVM no worse than single switching!
Quality of Optimization

DA  TSVM (1)  TSVM (max)

![Graph showing the quality of optimization for different methods.](image)

**Observations**
- DA often gives significantly better minimizers [aut-avn, ccat, ppcmac].
- Multiple switching TSVM no worse than single switching!
Quality of Optimization

Observations

- DA often gives significantly better minimizers [aut-avn, ccat, pcmac].
- Multiple switching TSVM no worse than single switching!
Generalization Performance

**Observations**
- Unlabeled data always very useful!
- TSVM’s worse minimizers also generalize fairly well.
- Max-switching performs as well as single switching.
Generalization Performance

Observations

- Unlabeled data always very useful!
- TSVM’s worse minimizers also generalize fairly well.
- Max-switching performs as well as single switching.
Observations

- Unlabeled data always very useful!
- TSVM’s worse minimizers also generalize fairly well.
- Max-switching performs as well as single switching.
### Speed comparison with SVM-Light

<table>
<thead>
<tr>
<th>Dataset</th>
<th>$\text{SVM}^\text{light}$</th>
<th>TSVM(1)</th>
<th>TSVM(max)</th>
<th>DA</th>
</tr>
</thead>
<tbody>
<tr>
<td>aut-avn</td>
<td>&gt; 1 day</td>
<td>1.6hrs</td>
<td>7min</td>
<td>24min</td>
</tr>
<tr>
<td>real-sim</td>
<td>&gt; 1 day</td>
<td>1.7hrs</td>
<td>6min</td>
<td>19min</td>
</tr>
<tr>
<td>ccat</td>
<td>4hrs</td>
<td>40min</td>
<td>6.5min</td>
<td>20min</td>
</tr>
<tr>
<td>gcat</td>
<td>&gt; 1 day</td>
<td>20min</td>
<td>6min</td>
<td>3min</td>
</tr>
<tr>
<td>33-36</td>
<td>14hrs</td>
<td>2hrs</td>
<td>5min</td>
<td>7min</td>
</tr>
<tr>
<td>pcmac</td>
<td>167sec</td>
<td>4sec</td>
<td>2sec</td>
<td>12sec</td>
</tr>
</tbody>
</table>

- Massive speedups over SVM-Light
- $\text{TSVM}(1) < \text{DA} < \text{TSVM (max)}$
- Implemented in Matlab, C much faster. Easily parallelizable.
### Larger-Scale Experiment

**Reuters C15: 804414 examples, 47256 features (r=0.18)**

<table>
<thead>
<tr>
<th>Method</th>
<th>l=100</th>
<th>l=1000</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM</td>
<td>74.59</td>
<td>84.79</td>
</tr>
<tr>
<td>TSVM</td>
<td>77.73</td>
<td>86.60</td>
</tr>
<tr>
<td>DA</td>
<td>78.11</td>
<td>85.79</td>
</tr>
<tr>
<td><strong>Obj. Value</strong></td>
<td><strong>TSVM</strong></td>
<td>0.094673</td>
</tr>
<tr>
<td></td>
<td><strong>DA</strong></td>
<td>0.08073</td>
</tr>
<tr>
<td><strong>CPU Time</strong></td>
<td><strong>TSVM</strong></td>
<td>1hr 22min (27670)</td>
</tr>
<tr>
<td></td>
<td><strong>DA</strong></td>
<td>2hr 8min</td>
</tr>
</tbody>
</table>
Summary of Experimental Results

- Unlabeled data is very useful.
- DA better optimizer than TSVM.
- Both compete well in terms of generalization.
- Local minima issues less severe on text than on other domains.
- Massive speedups over SVM-Light. Max-switching TSVM is fastest, DA comparable.
Outline

1. Fast (fully supervised) Linear SVMs
2. The cluster assumption for SSL
3. Semi-supervised SVMs
   - An objective function to implement cluster assumption
   - A Scalable Label-switching Algorithm
   - The Problem of Non-convexity
   - A Deterministic Annealing (DA) approach
4. Empirical Studies
5. Extensions
Extensions

- Handling uncertain class ratios.
- Better annealing sequence for DA.
- Fast $l_2$-SVMs can be used to implement other SSL assumptions: Manifold (Laplacian SVM) and Co-training (Co-regularization).
- Software implementation available: $\textbf{SVM}_{\text{lin}}$: Fast Linear SVM Solvers for Supervised and Semi-supervised Learning,
  http://www.cs.uchicago.edu/~vikass/svmlin.html