Large Scale Manifold Transduction

Michael Karlen†, Jason Weston*, Ayse Erkan‡ & Ronan Collobert*

* NEC Labs America, Princeton, USA
† École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland
‡ New York University, NY, USA
Methods of semi-supervised learning

Perhaps two most popular approaches:

(i) **Margin-based**: maximize margin on unlabeled data
   - i.e. TSVMs

(ii) **Manifold-based**: use graph structure to infer metric on examples
    - manifold learning
    - e.g. “spectral clustering” kernels, label propagation & LapSVMs

- All methods need lots of unlabeled data but are too slow
- Both approaches → decision rule lies in a region of low density.
- LDS (Low Density Separation)
  - two-stages: build kernel from graph → apply TSVM.
  - pros: better test error  cons: ad-hoc, slow
From the authors of LDS:

“We observe that the time (and to some degree, also space) complexities of all methods investigated here prohibit the application to really large sets of unlabeled data, say, more than a few thousand. Thus, work should also be devoted to improvements of the computational efficiency of algorithms, ideally of LDS.”

[Chapelle & Zien]
Summary of our Contribution

a) TSVM can be trained **online** by SGD = fast (1M examples...)  
b) TSVM loss applied to deep neural networks = powerful, nonlinear  
c) New generalization of TSVM loss = more robust  
   - *Uses graph / manifold information directly*  
   - *Generalizes TSVM + manifold learning into one loss*  
   - *Unified, fast version of LDS*
Existing Semi-Supervised Techniques: TSVM

SVM: \( \min_{w,b} \gamma \|w\|^2 + \sum_{i=1}^{L} H(y_i f(x_i)) \) 

\( H(x) = \max(0, 1 - x) \)

- TSVM [Vapnik]: push unlabeled data far from margin = clustered

\[ \text{SVM} + \lambda \sum_{i=1}^{U} H(|f(x_i^*)|) \]

\(+ balancing \ constraint\)
Existing TSVM implementations

pros: good objective func.
cons: hard to optimize (non-convex), so lots of implementations:

- **SVMLight-TSVM** - heuristic label swapping, retrain SVM
  \[ f(x) = \sum_{i=1}^{L} \alpha_i y_i K(x_i, x) + \sum_{i=1}^{U} \alpha_i^* K(x_i^*, x) + b \]
  balancing: \[ \frac{1}{U} \sum_{i=1}^{U} y_i^* = \frac{1}{L} \sum_{i=1}^{L} y_i \]

- **VS³VM** - concave-convex minimization: iterative LP

- **∇-TSVM** - linearize via KPCA, gradient descent
  balancing: \[ \frac{1}{U} \sum_{i=1}^{U} f(x_i^*) = \frac{1}{L} \sum_{i=1}^{L} y_i \]

- **CCCP-TSVM** - nonlinear generalization of VS³VM

- **Large Scale Linear TSVMs** - label swapping + fast linear SVMs
Existing Semi-Supervised Techniques

**SVM:**\[\min_{w, b} \gamma ||w||^2 + \sum_{i=1}^{L} H(y_i f(x_i))\]

- **TSVM [Vapnik]:** push unlabeled data far from margin \(=\) clustered
  \[\text{SVM} + \lambda \sum_{i=1}^{U} H(\|f(x_i^*)\|)\]
  + balancing constraint

- **LapSVM [Belkin et al.]:** unlabeled neighbors have same output
  \[\text{SVM} + \lambda \sum_{i,j=1}^{U} W_{ij} \|f(x_i^*) - f(x_j^*)\|^2\]
  e.g. \(W_{ij} = 1\) if two points are neighbors, 0 otherwise.

- **LDS [Chapelle et al.]:** Isomap features \(\rightarrow\) TSVM
We propose the following algorithm, Manifold Transduction:

\[
\text{minimize} \quad \frac{1}{L} \sum_{i=1}^{L} \ell(f(x_i), y_i) + \frac{\lambda}{U^2} \sum_{i,j=1}^{U} W_{ij} \ell(f(x_i^*), y^*(\{i, j\})) \\
\text{s.t. balancing constraint}
\]

where

\[
y^*(\{i, j\}) = \text{sign}(f(x_i^*) + f(x_j^*))
\]

- General case: \( \ell(f(x_i^*), y^*(N)), \ y^*(N) = \arg\max \sum_{k \in N} f(x_k^*) \).
- If \( W_{ii} = 1, W_{ij} = 0 \) for \( i \neq j \) \( \rightarrow \) recover TSVM

We now discuss the choice of:
(a) model, (b) balancing constraint, (c) optimization strategy.
Model: NNs or CNNs

Linear case: *same as other methods.*
Nonlinear case: *use neural nets rather than kernels.*

**NN:**

- very high level representation:
  - MAN
  - SITTING

  ... etc ...

- slightly higher level representation

raw input vector representation:

\[ \mathbf{x} = [23, 19, 20, 18] \]

\[ x_1, x_2, x_3, \ldots, x_n \]

\[ f_{NN}(x) \text{ faster to calculate than } f_{SVM}(x) = \sum_{i=1}^{L} \alpha_i y_i K(x_i, x) + \sum_{i=1}^{U} \alpha_i^* K(x_i^*, x) \]
Online Balancing constraint: methods

Two online methods:

- \( \nabla \text{bal} \): gradient step to ensure \( \frac{1}{U} \sum_{i=1}^{U} f(x_i^\ast) = (p_{\text{est}}(y = 1) - p_{\text{est}}(y = -1)) \)

- \( \text{ignore-bal} \): IF fraction of recent assignments to class \( y^\ast \) < \( p_{\text{est}}(y^\ast) \) THEN Make a gradient step

\( p_{\text{est}}(y = Y) \) is the prediction of probability of label \( y \)

- \( p_{\text{trn}} \) - use training set distribution

- \( p_{k\text{nn}} \) - predict labels of \( k \)-nn, use label distribution

- \( p_{\text{tst}} \) - use test set distribution (cheat)
Online Manifold Transduction

**Input:** labeled data \((x_i, y_i)\) and unlabeled data \(x_i^*\)

repeat

- Pick a random labeled example \((x_i, y_i)\)
- Make a gradient step to optimize \(\ell(f(x_i), y_i)\)
- Pick a random unlabeled example \(x_i^*\)
- Pick a random neighbor \(x_j^*\) of \(x_i^*\)
- Predict label \(y^* = y^*(\{i, j\})\)
- if fraction of recent assignments to class \(y^* < p_{est}(y^*)\) then
  - Make a gradient step for \(\ell(f(x_i^*), y^*)\)
end if

until stopping criteria is met.

- \(f(x)\) is as deep a network as you want!

- *Vanilla Transduction:* use \(y^* = f(x_i)\)
## Semi-Supervised Experiments

Typical *semi-supervised* datasets:

<table>
<thead>
<tr>
<th>data set</th>
<th>classes</th>
<th>dims</th>
<th>points</th>
<th>labeled</th>
</tr>
</thead>
<tbody>
<tr>
<td>g50c</td>
<td>2</td>
<td>50</td>
<td>500</td>
<td>50</td>
</tr>
<tr>
<td>Text</td>
<td>2</td>
<td>7511</td>
<td>1946</td>
<td>50</td>
</tr>
<tr>
<td>Uspst</td>
<td>10</td>
<td>256</td>
<td>2007</td>
<td>50</td>
</tr>
<tr>
<td>Mnist1h</td>
<td>10</td>
<td>784</td>
<td>70k</td>
<td>100</td>
</tr>
<tr>
<td>Mnist1k</td>
<td>10</td>
<td>784</td>
<td>70k</td>
<td>1000</td>
</tr>
<tr>
<td>Mnist1k+Invar</td>
<td>10</td>
<td>784</td>
<td>630k</td>
<td>1000</td>
</tr>
</tbody>
</table>
# Deep Semi-Supervised Results

<table>
<thead>
<tr>
<th>Method</th>
<th>g50c</th>
<th>Text</th>
<th>Uspst</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM</td>
<td>8.32</td>
<td>18.86</td>
<td>23.18</td>
</tr>
<tr>
<td>SVMLight-TSVM</td>
<td>6.87</td>
<td>7.44</td>
<td>26.46</td>
</tr>
<tr>
<td>CCCP-TSVM</td>
<td>5.62</td>
<td>7.97</td>
<td>16.57</td>
</tr>
<tr>
<td>∇TSVM</td>
<td>5.80</td>
<td>5.71</td>
<td>17.61</td>
</tr>
<tr>
<td>LapSVM*</td>
<td>5.4</td>
<td>10.4</td>
<td>12.7</td>
</tr>
<tr>
<td>LDS*</td>
<td>5.4</td>
<td>5.1</td>
<td>15.8</td>
</tr>
<tr>
<td>Label propagation</td>
<td>17.30</td>
<td>11.71</td>
<td>21.30</td>
</tr>
<tr>
<td>graph</td>
<td>8.32</td>
<td>10.48</td>
<td>16.92</td>
</tr>
<tr>
<td>NN</td>
<td>8.54</td>
<td>15.87</td>
<td>24.57</td>
</tr>
<tr>
<td>TNN</td>
<td>6.34</td>
<td>6.11</td>
<td>16.06</td>
</tr>
<tr>
<td>ManTNN</td>
<td>5.66</td>
<td>5.34</td>
<td>11.90</td>
</tr>
</tbody>
</table>
## Online Balancing constraint: experiments

<table>
<thead>
<tr>
<th></th>
<th>Uspst</th>
<th>g50c</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><em>p_{trn}</em></td>
<td><em>p_{knn}</em></td>
</tr>
<tr>
<td><strong>TNN</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>no bal</td>
<td>22.3</td>
<td>–</td>
</tr>
<tr>
<td>\nabla bal</td>
<td>30.4</td>
<td>29.3</td>
</tr>
<tr>
<td>ignore-bal</td>
<td>19.1</td>
<td>16.1</td>
</tr>
<tr>
<td><strong>ManTNN</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ignore-bal</td>
<td>15.6</td>
<td>11.9</td>
</tr>
</tbody>
</table>
# Deep Semi-Supervised MNIST

<table>
<thead>
<tr>
<th>Model</th>
<th>Mnist1h</th>
<th>Mnist1k</th>
<th>Mnist1k+Invar</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM</td>
<td>23.44</td>
<td>7.77</td>
<td></td>
</tr>
<tr>
<td>CCCP-TSVM</td>
<td>16.81</td>
<td>5.38</td>
<td></td>
</tr>
<tr>
<td>NN</td>
<td>25.81</td>
<td>10.70</td>
<td></td>
</tr>
<tr>
<td>TNN</td>
<td>18.02</td>
<td>6.66</td>
<td>5.23</td>
</tr>
<tr>
<td>ManTNN</td>
<td>7.30</td>
<td>2.88</td>
<td>2.43</td>
</tr>
<tr>
<td>CNN</td>
<td>22.98</td>
<td>6.45</td>
<td></td>
</tr>
<tr>
<td>TCNN</td>
<td>13.01</td>
<td>3.50</td>
<td></td>
</tr>
<tr>
<td>ManTCNN</td>
<td>6.65</td>
<td>2.15</td>
<td></td>
</tr>
<tr>
<td>ManTCNN (p_{tst})</td>
<td>1.96</td>
<td>1.87</td>
<td></td>
</tr>
</tbody>
</table>
Timing results

Mnist1h or 1k: CCCP-TSVMs take $\sim 42$ hours on the same machine.

Nonlinear TNN (200 HUs) process 1M unlab. examples in 12.5 mins.

Mnist1k+Invar: TNN and ManTNN take $\sim 4$hrs.
• Large-scale, online nonlinear Transduction.

• Combines two main principles for SSL: transduction + graph-based regularization.

• Many variants of $y^*_N$ - nearest neighbors, body+link (co-training), averaging classifiers . . .