Large Scale Manifold Transduction

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

∗ NEC Labs America, Princeton, USA

† Ećole 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 $\rightarrow$ decision rule lies in a region of low density.
- LDS (Low Density Separation)
  two-stages: build kernel from graph $\rightarrow$ 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: $\min_{w,b} \gamma ||w||^2 + \sum_{i=1}^{L} H(y_i f(x_i))$ $\qquad H(x) = max(0, 1-x)$

- 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

$$f(x) = \sum_{i=1}^{L} \alpha_i y_i K(x_i, x) + \sum_{i=1}^{U} \alpha_i^* K(x_i^*, x) + b \qquad \text{balancing: } \frac{1}{U}\sum_{i=1}^{U} y_i^* = \frac{1}{L}\sum_{i=1}^{L} y_i$$

- VS^{3}VM - concave-convex minimization: iterative LP
- ∇-TSVM - linearize via KPCA, gradient descent

$$\text{balancing: } \frac{1}{U}\sum_{i=1}^{U} f(x_i^*) = \frac{1}{L}\sum_{i=1}^{L} y_i \,.$$

- CCCP-TSVM - nonlinear generalization of VS^{3}VM
- 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

Proposed Approach : Manifold Transduction

We propose the following algorithm, Manifold Transduction:

$$\text{minimize} \quad \frac{1}{L}\sum_{i=1}^{L} \ell(f(x_i), y_i) \; + \; \frac{\lambda}{U^2}\sum_{i,j=1}^{U} W_{ij} \; \ell\big(f(x_i^*), y^*(\{i,j\})\big)$$

s.t. balancing constraint

where

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

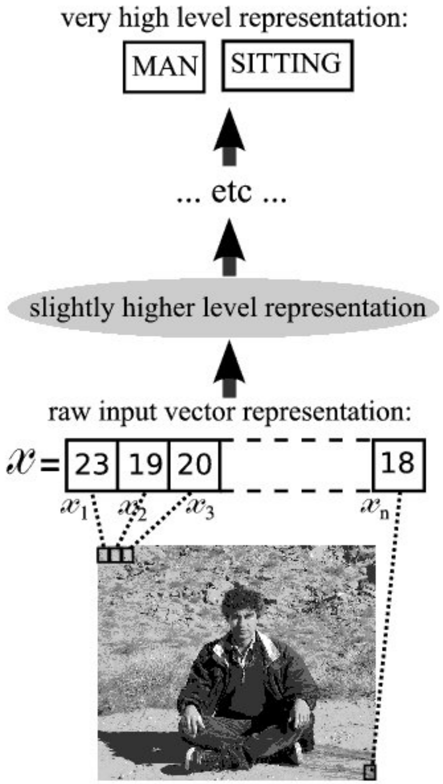
- General case: $\ell\big(f(x_i^*), y^*(N)\big)$, $\quad y^*(N) = \text{argmax} \sum_{k \in N} f(x_k^*)$.
- If $W_{ii} = 1, W_{ij} = 0$ for $i \neq j$ $\rightarrow$ recover TSVM

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

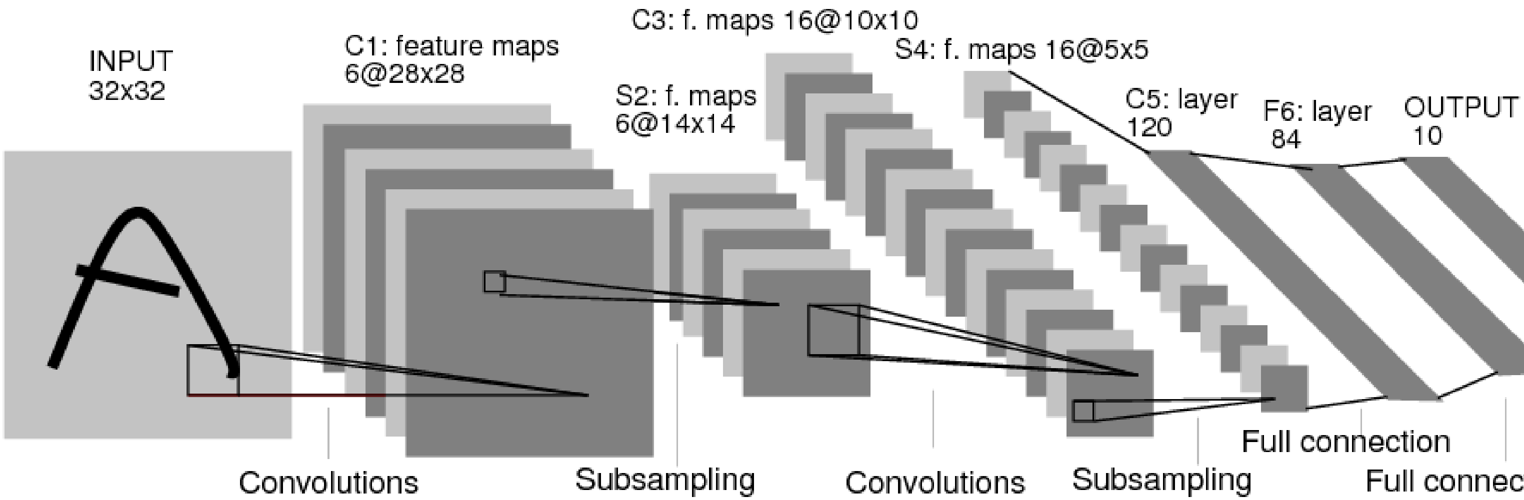
Model: NNs or CNNs

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

NN:



CNN:



$f_{NN}(x)$ 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:

- ∇bal: gradient step to ensure $\frac{1}{U}\sum_{i=1}^{U} f(x_i^*) = (p_{est}(y=1) - p_{est}(y=-1))$
- ignore-bal: IF fraction of recent assignments to class $y^* < p_{est}(y^*)$ THEN Make a gradient step

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

- p_{trn} - use training set distribution
- p_{knn} - predict labels of k-nn, use label distribution
- p_{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:

data set	classes	dims	points	labeled
g50c	2	50	500	50
Text	2	7511	1946	50
Uspst	10	256	2007	50
Mnist1h	10	784	70k	100
Mnist1k	10	784	70k	1000
Mnist1k+Invar	10	784	630k	1000

Deep Semi-Supervised Results

	g50c	Text	Uspst
SVM	8.32	18.86	23.18
SVMLight-TSVM	6.87	7.44	26.46
CCCP-TSVM	5.62	7.97	16.57
∇TSVM	5.80	5.71	17.61
LapSVM*	5.4	10.4	12.7
LDS*	5.4	5.1	15.8
Label propagation	17.30	11.71	21.30
graph	8.32	10.48	16.92
NN	8.54	15.87	24.57
TNN	6.34	6.11	16.06
ManTNN	5.66	5.34	11.90

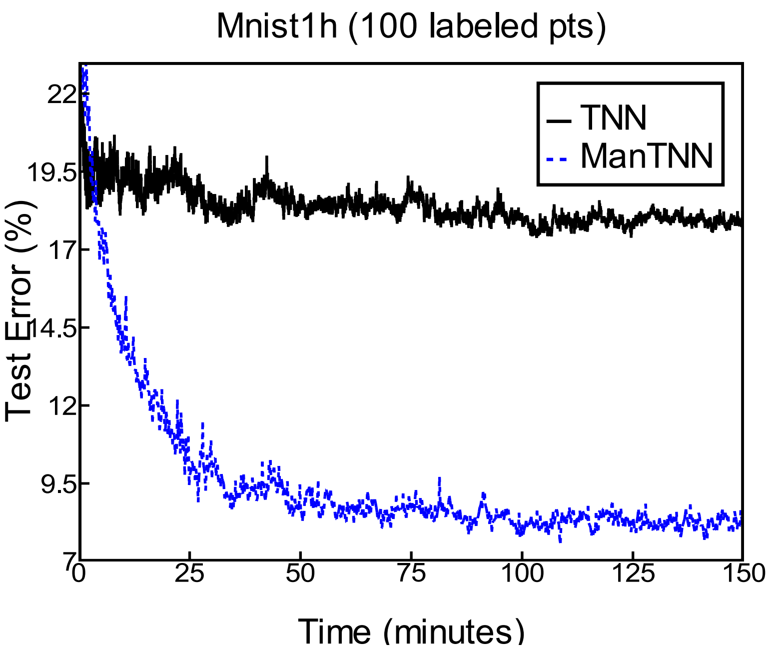
Online Balancing constraint: experiments

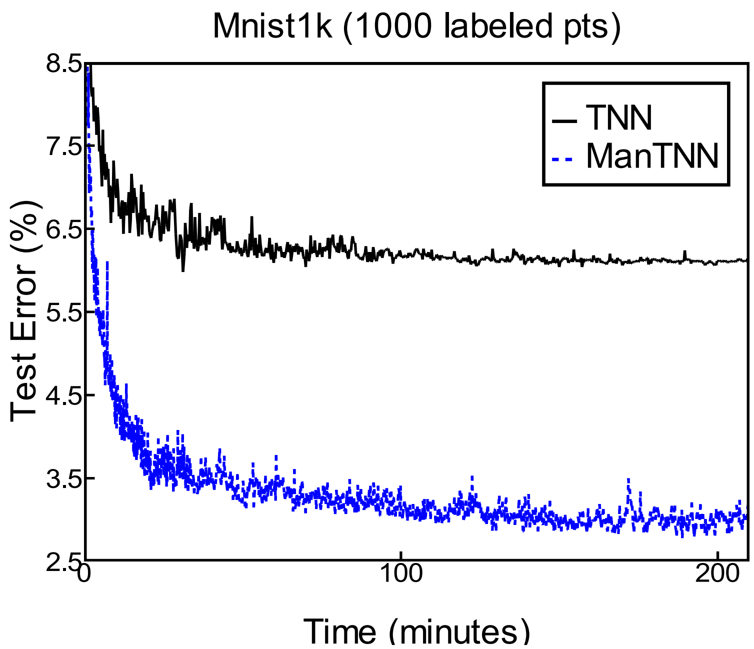
	Uspst			g50c		
	p_{trn}	p_{knn}	p_{tst}	p_{trn}	p_{knn}	p_{tst}
TNN						
no bal	22.3	–	–	6.5	–	–
∇bal	30.4	29.3	29.4	6.5	6.5	6.5
ignore-bal	19.1	16.1	12.5	6.1	6.3	6.3
ManTNN						
ignore-bal	15.6	11.9	8.5	5.9	5.7	5.5

Deep Semi-Supervised MNIST

	Mnist1h	Mnist1k	Mnist1k+Invar
SVM	23.44	7.77	
CCCP-TSVM	16.81	5.38	
NN	25.81	10.70	
TNN	18.02	6.66	5.23
ManTNN	7.30	2.88	2.43
CNN	22.98	6.45	
TCNN	13.01	3.50	
ManTCNN	6.65	2.15	
ManTCNN (p_{tst})	*1.96*	*1.87*	

Timing results





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

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

Mnist1k+Invar: TNN and ManTNN take ∼4hrs.

Conclusion

- 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 . . .*