Online Manifold Regularization: A New Learning Setting and Empirical Study

Andrew B. Goldberg[1], Ming Li[2], Xiaojin Zhu[1]

[1] Department of Computer Sciences, University of Wisconsin–Madison, USA
[2] National Key Laboratory for Novel Software Technology, Nanjing University, China

THE UNIVERSITY of WISCONSIN MADISON

Motivating Examples

Consider a mobile robot continuously learning to recognize interesting objects (x) with limited feedback from humans (y):

x_1 x_2 ... x_{1000} ... $x_{1000000}$ ...

$y_1 = 0$ n/a ... $y_{1000} = 1$ ... $y_{1000000} = 0$ ...

This is how children learn, too:

x_1 x_2 ... x_{1000} ... $x_{1000000}$ ...

$y_1 = 0$ n/a ... $y_{1000} = 1$ ... $y_{1000000} = 0$ ...

Unlike standard supervised learning:

- $n \to \infty$ examples arrive sequentially
- Cannot even store them all
- Most examples are unlabeled
- No iid assumption; $p(x, y)$ can change

New Paradigm: Online Semi-Supervised Learning

Main contribution: Merging settings

1. Online: learn from non-iid sequence, but fully labeled data
2. Semi-supervised: learn from iid batch, but (mostly) unlabeled data

Learning proceeds iteratively:

1. At time t, adversary picks $x_t \in \mathcal{X}, y_t \in \mathcal{Y}$ not necessarily iid; shows x_t to learner
2. Learner has $f_t : \mathcal{X} \mapsto \mathbb{R}$; predicts $f_t(x_t)$
3. With small probability, adversary reveals y_t; otherwise it abstains (unlabeled)
4. Learner updates to f_{t+1} based on x_t and y_t (if given). Repeat.

Review: Batch Manifold Regularization

A form of graph-based semi-supervised learning [Belkin et al. JMLR06]:

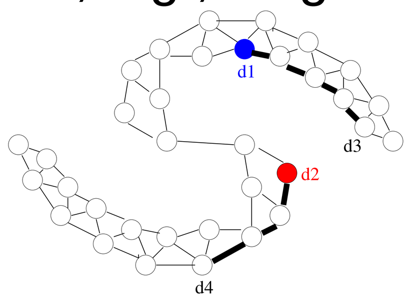
- Graph on $x_1 \dots x_n$
- Edge weights w_{st} encode similarity
- Assumption: similar x's have similar labels

Manifold regularization minimizes risk:

$$J(f) = \frac{1}{l}\sum_{t=1}^{T}\delta(y_t)c(f(x_t), y_t) + \frac{\lambda_1}{2}\|f\|_K^2 + \frac{\lambda_2}{2T}\sum_{s,t=1}^{T}(f(x_s) - f(x_t))^2 w_{st}$$

$c(f(x), y)$ convex loss function, e.g., hinge

Generalizes graph mincut and label propagation.



From Batch to Online

Batch risk = average instantaneous risks

$$J(f) = \frac{1}{T}\sum_{t=1}^{T} J_t(f)$$

Instantaneous risk

$$J_t(f) = \frac{T}{l}\delta(y_t)c(f(x_t), y_t) + \frac{\lambda_1}{2}\|f\|_K^2 + \lambda_2\sum_{i=1}^{t}(f(x_i) - f(x_t))^2 w_{it}$$

(includes graph edges between x_t and all previous x's)

Online Convex Programming

Instead of minimizing convex $J(f)$, reduce convex $J_t(f)$ at each step t

$$f_{t+1} = f_t - \eta_t \left.\frac{\partial J_t(f)}{\partial f}\right|_{f_t}$$

Remarkable no regret guarantee against adversary: [Zinkevich ICML03]
$\limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} J_t(f_t) - J(f^*) \le 0$

If no adversary (iid), the average classifier $\bar{f} = 1/T \sum_{t=1}^{T} f_t$ is good: $J(\bar{f}) \to J(f^*)$

Kernelized Algorithm

New representation: $f_t(\cdot) = \sum_{i=1}^{t-1}\alpha_i^{(t)}K(x_i, \cdot)$

- Init: $t = 1, f_1 = 0$
- Repeat
  1. Receive x_t, predict $f_t(x_t) = \sum_{i=1}^{t-1}\alpha_i^{(t)}K(x_i, x_t)$
  2. Occasionally receive y_t
  3. Update f_t to f_{t+1} by adjusting coefficients

$$\alpha_i^{(t+1)} = (1-\eta_t\lambda_1)\alpha_i^{(t)} - 2\eta_t\lambda_2(f_t(x_i) - f_t(x_t))w_{it}, \quad i < t$$
$$\alpha_i^{(t+1)} = 2\eta_t\lambda_2\sum_{i=1}^{t}(f_t(x_i) - f_t(x_t))w_{it} - \eta_t\frac{T}{l}\delta(y_t)c'(f(x_t), y_t) \quad i = t$$

  4. Store x_t, let $t = t + 1$

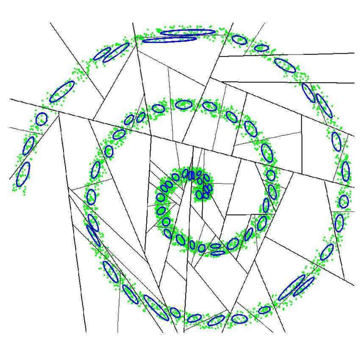
Sparse Approximations

The algorithm is impractical

- Space $O(T)$: stores all previous examples
- Time $O(T^2)$: each new example compared to all previous ones
- In reality, $T \to \infty$ for life-long learning

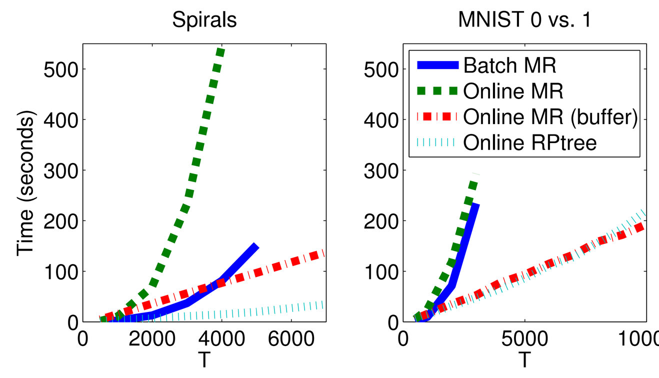
Two ways to speed up:

- Buffering
  - Keep a size τ buffer
  - Approximate representers: $f_t = \sum_{i=t-\tau}^{t-1}\alpha_i^{(t)}K(x_i, \cdot)$
  - Approximate instantaneous risk; only τ edge terms
  - Dynamic graph on examples in the buffer
- Random projection tree
  - Discretize data manifold by online clustering using RP tree [Dasgupta and Freund, STOC08]
  - Use clusters as representers
  - Approximate risk using "cluster graph"



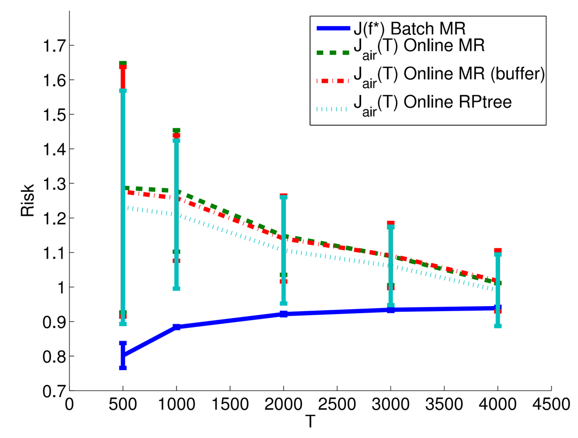
Experiment: Runtime

Buffering and random projection tree scale linearly, enabling life-long learning



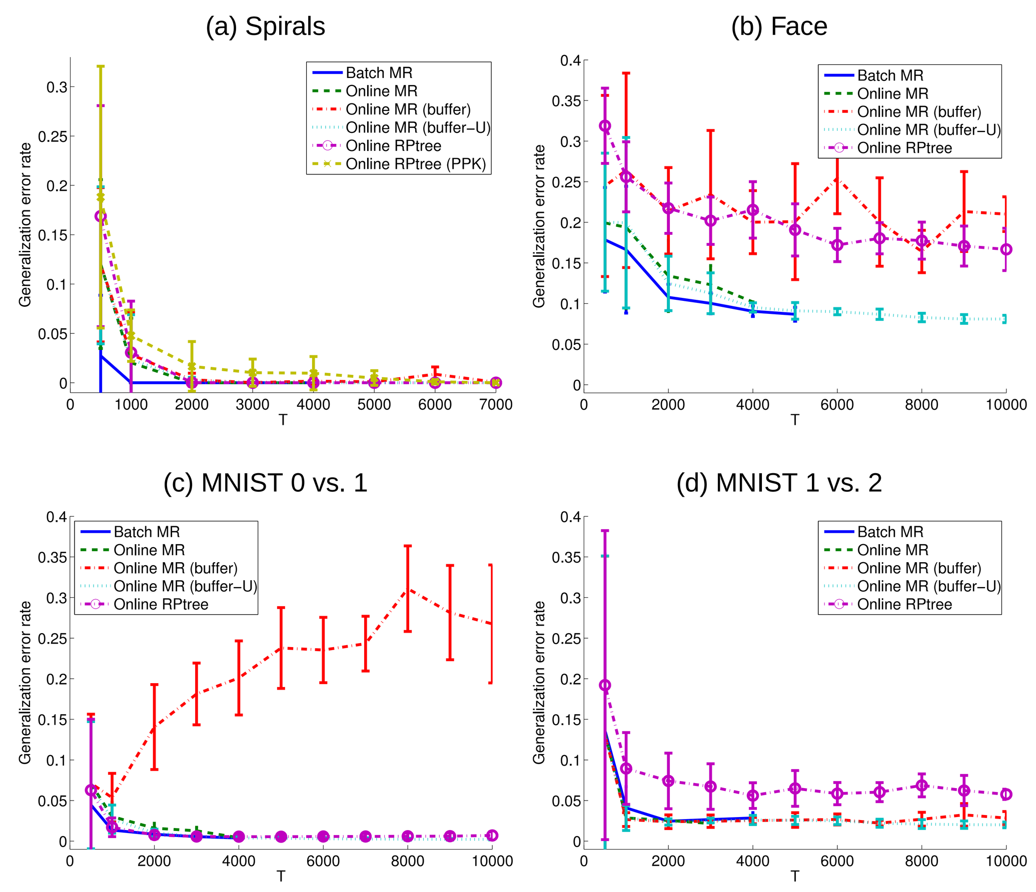
Experiment: Risk

Online MR risk $J_{air}(T) \equiv \frac{1}{T}\sum_{t=1}^{T} J_t(f_t)$ approaches batch risk $J(f^*)$ as T increases



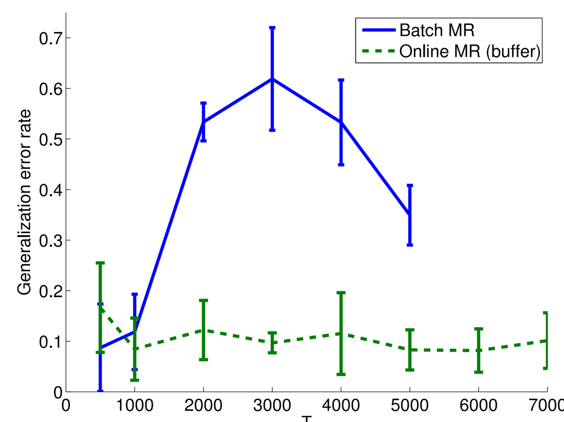
Experiment: Generalization Error of $\bar{f}$ if IID

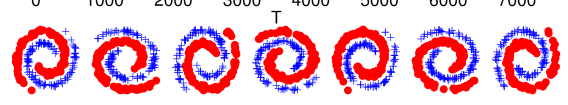
Variation of buffering as good as batch MR (prefer to keep labeled examples in buffer)



Experiment: Concept Drift

- Slowly rotating spirals; both $p(x)$ and $p(y|x)$ change over time
- Test set ~ current $p(x, y)$ at time T
- Online MR buffering f_T beats batch f^*





Summary

- Introduced online semi-supervised learning framework and specialization for MR
- Sparse approximations to make it practical: buffering and random projection tree
- Future work: new bounds, new algorithms (e.g., S3VM, multi-view)

http://pages.cs.wisc.edu/~goldberg/publications.html goldberg@cs.wisc.edu