Learning with Local and Global Consistency

Dengyong Zhou, Olivier Bousquet,
Thomas Navin Lal, Jason Weston, and
Bernhard Scholkopf

8 April 2014

Presented by: Wittawat Jitkrittum

Gatsby Tea Talk

About This Talk

- Zoltan's talk 3 weeks ago:
  - Wasserstein Propagation for Semi-Supervised Learning
- The term "label propagation" is used often in semi-supervised learning.
- What is its origin ? Seems to be . . . (I think)
  - Learning with Local and Global Consistency. NIPS 2003 ([Zhou et al., 2003]).

Outline

Outline

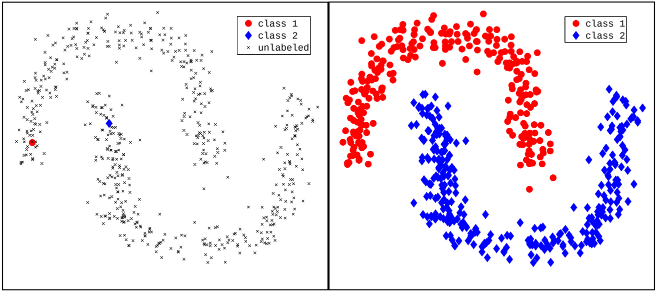
Transduction

- **Input**: $\overbrace{\{x_i, y_i\}_{i=1}^{l}}^{l \text{ labeled points}}$ and $\overbrace{\{x_i\}_{i=l+1}^{l+u}}^{u \text{ unlabeled points}}$
  - Infer just $\{y_i\}_{i=l+1}^{l+u}$, not the mapping $f : X \mapsto Y$.
  - Assume $l \ll u$.
  - $n = l + u$
  - $y_i \in \{1, \ldots C\}$ (classification task)
- An easier problem than induction (i.e., learning f).
- Label propagation does just that.
- Application: document categorization

Outline

What Is Label Propagation ?

- Use $\{x_i, y_i\}_{i=1}^{l}$ (small l) and $\{x_i\}_{i=l+1}^{l+u}$ (large u) to find $\{y_i\}_{i=l+1}^{l+u}$.
- $\Rightarrow$ go from left plot to right plot



- **Idea**: Each point spreads label information to its neighbors
- Neighborhood defined by similarity matrix W.

Set Up

- For each x_i, define

$$Y_i := (\delta(y_i = 1), \ldots, \delta(y_i = C)) \in \{0,1\}^{1\times C}.$$

If x_i is unlabeled i.e., $i \geq l+1$, then $Y_i = \mathbf{0}_{1\times C}$.

- For each x_i, label propagation finds a nonnegative scoring vector $F_i \in \mathbb{R}_+^{1\times C}$.
  - $F_i = (f_{i1}, \ldots, f_{iC})$ = class membership scores
- Label propagation finds $F = \begin{pmatrix} F_1 \\ \vdots \\ F_{l+u} \end{pmatrix}$ given $Y = \begin{pmatrix} Y_1 \\ \vdots \\ Y_{l+u} \end{pmatrix}$.
- Y is fixed.

Label Propagation Algorithm

1. Form an affinity (similarity) matrix $W \in \mathbb{R}^{n \times n}$. Set $W_{ii} = 0$.
2. Normalize W by
$$S = D^{-1/2} W D^{-1/2}$$
where D is diagonal with $D_{ii} = \sum_j W_{ij}$.
3. Iterate
$$F(t+1) \leftarrow \alpha S F(t) + (1 - \alpha) Y$$
where $\alpha \in (0, 1)$ and $F(0) = Y$.
4. Label x_i with
$$y_i = \arg\max_k F^*_{i,k}$$
where $F^* := \lim_{t \to \infty} F(t)$.

Affinity Matrix Construction

Various choices from ([Belkin and Niyogi, 2003])

- ϵ-neighborhoods:

$$W_{ij} = 1 \text{ if } \|x_i - x_j\|^2 < \epsilon$$

  May lead to several connected components

- k nearest neighbors (kNN)

$$W_{ij} = 1 \text{ if } x_i \in \mathrm{kNN}(x_j) \text{ or } x_j \in \mathrm{kNN}(x_i)$$

- Gaussian kernel: $W_{ij} = \exp\left(-\|x_i - x_j\|^2 / 2\sigma^2\right)$

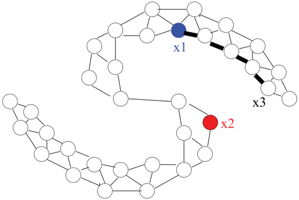


Image from [Zhu, 2007]

Notes on Label Propagation

- W captures the intrinsic structure of the data.
- Set $W_{i,i} = 0$ to avoid self-reinforcement.
- α trade-offs information from neighbors and Y

$$F(t+1) \leftarrow \alpha S F(t) + (1-\alpha) Y$$

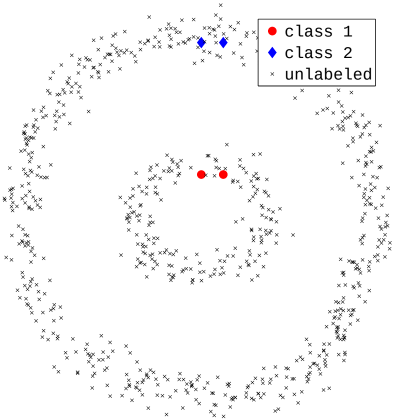
  High $\alpha \Rightarrow$ trust neighbors ($\alpha = 0.99$ in the paper)
- Analytic update

$$F^* = (1-\alpha) \left(I_{n\times n} - \alpha S\right)^{-1} Y$$

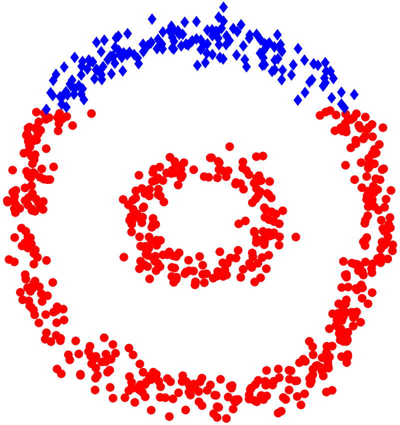
  (independent of $F(0)$)

Label Propagation on 2circs Data

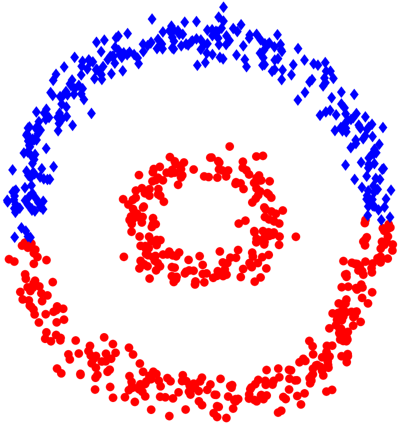
- Affinity matrix W is constructed with Gaussian kernel with small width



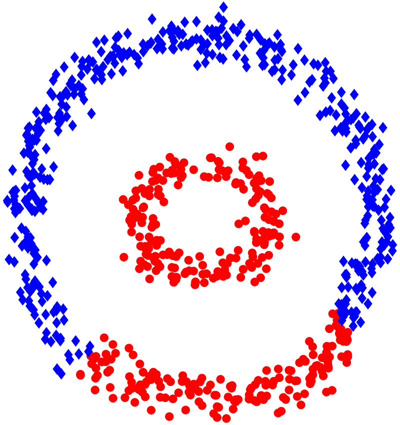
After 1 Iteration



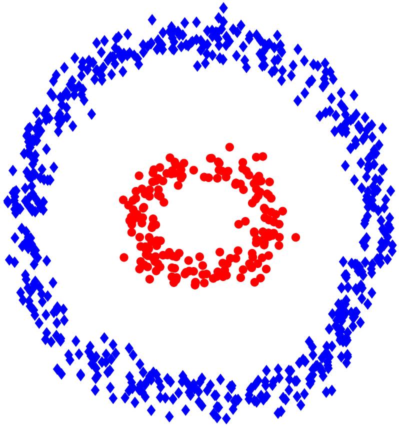
After 10 Iterations



After 40 Iterations



After 80 Iterations (converged)



Outline

Regularization Framework

- $F^* = \arg\min_F Q(F)$ (loss function) where

$$Q(F) = \frac{1}{2}\left(\underbrace{\sum_{i=1}^{n}\sum_{j=1}^{n} W_{i,j} \left\| \frac{F_i}{\sqrt{D_{i,i}}} - \frac{F_j}{\sqrt{D_{j,j}}} \right\|^2}_{\text{smoothness constraint}} + \mu \underbrace{\sum_{i=1}^{n} \|F_i - Y_i\|^2}_{\text{fitting constraint}} \right)$$

- **Implication:** A good F should
  - not change too much between nearby points (smoothness)
  - not change too much from the initial label assignment Y (fitting constraint)
- Trade-off captured by μ (regularization parameter).

Solve $Q(F)$

- Rewrite $Q(F)$,

$$\begin{aligned} Q(F) &= \operatorname{tr}\left(F^\top (I-S) F\right) + \\ &\quad \frac{\mu}{2}\left[\operatorname{tr}\left(FF^\top\right) - 2\operatorname{tr}\left(FY^\top\right) + \operatorname{tr}\left(YY^\top\right)\right] \end{aligned}$$

- Differentiate w.r.t. F

$$\begin{aligned} \frac{\partial Q}{\partial F} &= 2(I-S)F + \mu(F-Y) = \mathbf{0} \\ F^* &= (\mu I - 2S)^{-1} Y \end{aligned}$$

- Recall previously $F^* = (1-\alpha)(I-\alpha S)^{-1} Y$.
- Equivalent solution with $\mu \propto 1/\alpha$.

Why Normalize W ?

$$S = D^{-1/2} W D^{-1/2}$$

- Eigenvalues of S in $[-1, 1]$. Necessary for the convergence.
- Eigen-decompose $S = VCV^\top$.

$$\begin{aligned} C &= V^\top D^{-1/2} W \overbrace{D^{-1/2} V}^{A} \\ &= V^\top D^{1/2} D^{-1} D^{1/2} V \end{aligned}$$

Since $A^{-1} = V^\top D^{1/2}$ (V orthogonal),

$$\begin{aligned} C &= A^{-1} D^{-1} W A \\ \Rightarrow D^{-1} W &= A C A^{-1} \end{aligned}$$

- C contains eigenvalues of $D^{-1}W$.
- $D^{-1}W$ is a stochastic matrix. Rows sum to 1.
  - Eigenvalues $|C_{ii}| \leq 1$.

Convergence

$$F(t+1) \leftarrow \alpha S F(t) + (1-\alpha) Y$$

$$F(t) = (\alpha S)^{t-1} Y + (1-\alpha) \sum_{i=0}^{t-1} (\alpha S)^i Y$$

Take the limit

$$\begin{aligned}
F^* &= \lim_{t\to\infty} F(t) = \overbrace{\lim_{t\to\infty} (\alpha S)^{t-1}}^{\mathbf{0}} Y + (1-\alpha) \overbrace{\lim_{t\to\infty} \sum_{i=0}^{t-1} (\alpha S)^i}^{B} Y \\
B &= I + \alpha S + (\alpha S)^2 + \cdots \text{(convergent series)} \\
\alpha S B &= \alpha S + (\alpha S)^2 + \cdots \\
B - \alpha S B &= I \\
\Rightarrow B &= (I - \alpha S)^{-1}
\end{aligned}$$

Substitute B back: $F^* = (1-\alpha)(I - \alpha S)^{-1} Y$

Outline

Conclusions

- Transduction is a task to predict labels of the observed unlabeled points.
- No mapping function $f : X \mapsto Y$ is learned.
- Label propagation tries to generate smooth outputs w.r.t. W
- Analytic solution.

References I

Belkin, M. and Niyogi, P. (2003).
Laplacian eigenmaps for dimensionality reduction and data representation.
Neural Computation, 15:1373–1396.

Belkin, M., Niyogi, P., and Sindhwani, V. (2005).
On manifold regularization.

Zhou, D., Bousquet, O., Lal, T. N., Weston, J., and Schölkopf, B. (2003).
Learning with local and global consistency.
In NIPS.

Zhu, X. (2007).
Semi-supervised learning tutorial.

Learning Paradigms

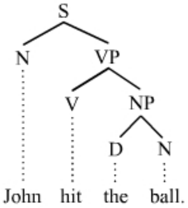
- Supervised learning
  - $\{(x_i, y_i)\}_{i=1}^n \Rightarrow$ Infer the mapping $f : X \mapsto Y$
  - Regression when $Y \in \mathbb{R}$. Classification when $Y \in \{1, \dots C\}$.
- Unsupervised learning
  - $\{x_i\}_{i=1}^n \Rightarrow$ Find hidden structure in the data
  - In clustering, find $y_i \in \{1, \dots, C\}$ (labels) such that $\{x_i\}_i$ with the same label are "similar".
- Semi-supervised learning
  - l of $\{x_i, y_i\}_{i=1}^l$ (labeled) and u of $\{x_i\}_{i=l+1}^n$ (unlabeled) $\Rightarrow$ Infer the mapping $f : X \mapsto Y$ (inductive).
  - $n = l + u$. Usually $l \ll u$.
- Reinforcement learning

Motivations for Semi-Supervised Learning

- Example task: web categorization
  - x_i = a web page
  - y_i = category
  - Goal: learn f : web page $\mapsto$ category
- Manual page annotation is time-consuming.
- Abundance of unlabeled sentences.
- Ideally, use both labeled and unlabeled data to build a better learner.

Motivations for Semi-Supervised Learning

- Example task: natural language parsing ([Zhu, 2007]).
  - x_i = sentence
  - y_i = parse tree
  - Goal: learn f : sentence $\mapsto$ parse tree



- Manual parse tree annotation is time-consuming.
- Abundance of unlabeled sentences.
- Ideally, use both labeled and unlabeled data to build a better learner.

How can unlabeled data help ?

Example from [Belkin et al., 2005].

- 2 classes $(C = 2)$. 2 labeled points. $\{(x_1, \text{blue}), (x_2, \text{red})\}$

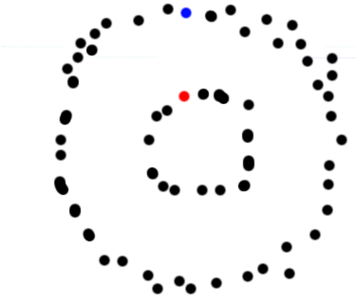
How can unlabeled data help ?

Example from [Belkin et al., 2005].

- Best decision boundary

How can unlabeled data help ?

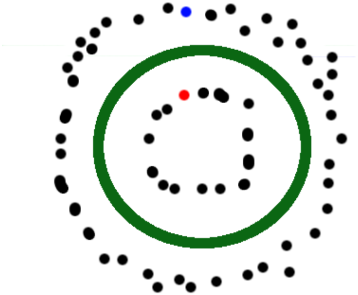
Example from [Belkin et al., 2005].



- $\{(x_1, \text{blue}), (x_2, \text{red})\}$ and $\{x_i\}_{i=3}^n$ (in black). Same decision boundary ?

How can unlabeled data help ?

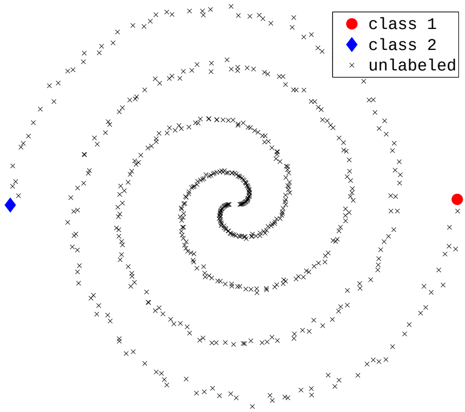
Example from [Belkin et al., 2005].



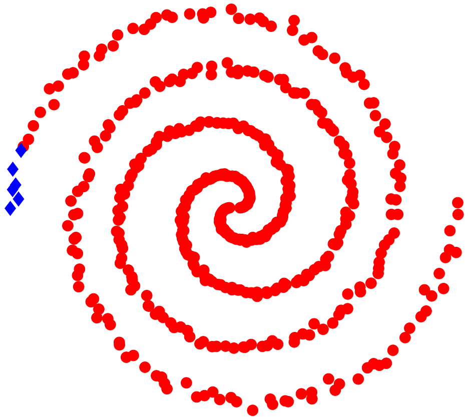
- So, unlabeled data can be helpful.

Label Propagation on 2spirals Data

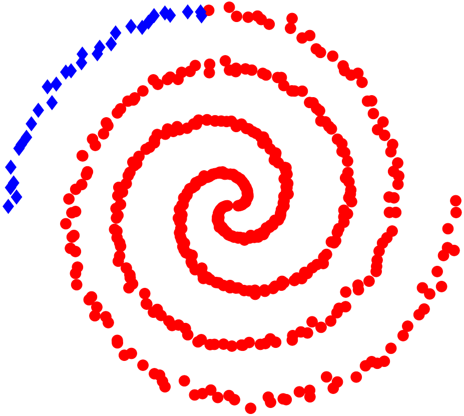
- Affinity matrix W is constructed with 5-NN.



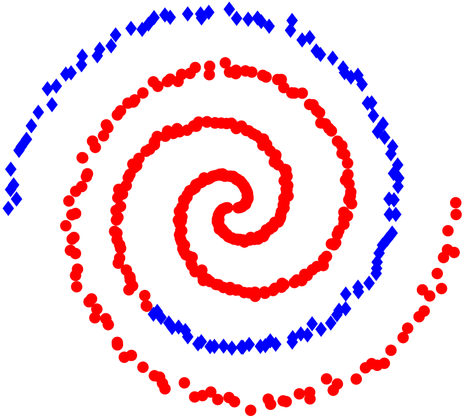
After 1 Iteration



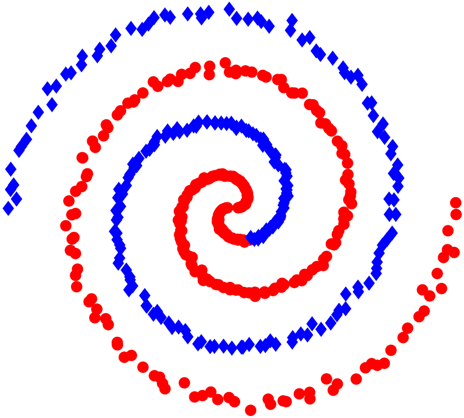
After 10 Iterations



After 40 Iterations



After 80 Iterations



After 100 Iterations (converged)