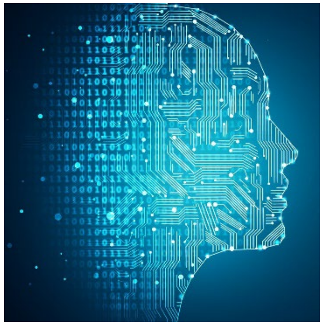
Machine Learning

Semi-supervised Learning

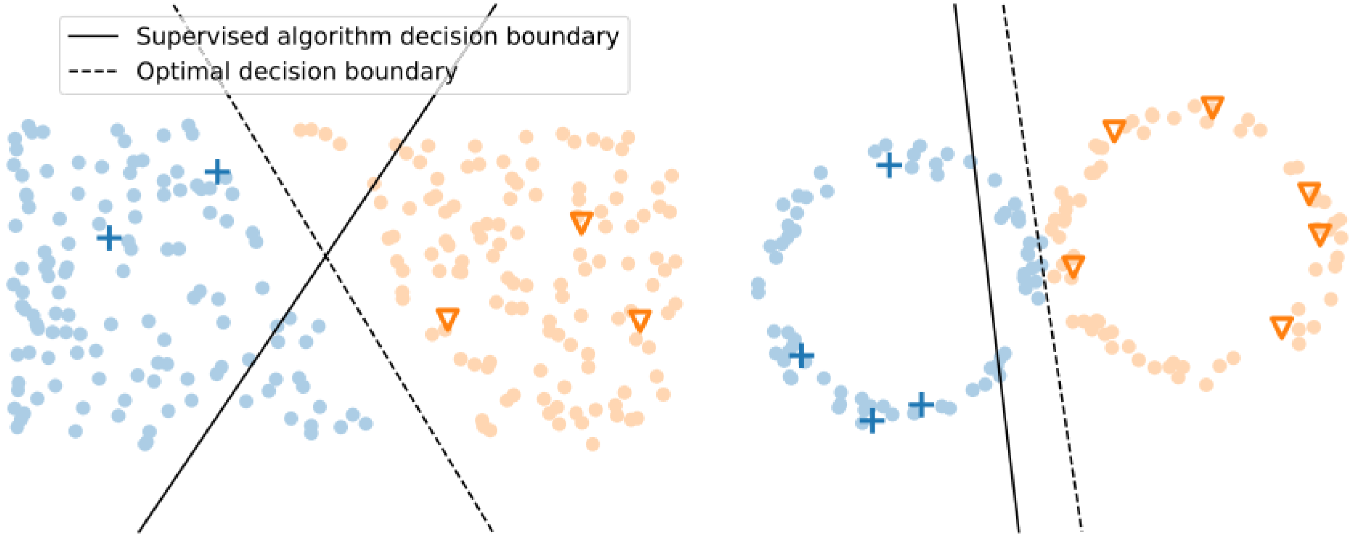
Kevin Moon (kevin.moon@usu.edu)

STAT/CS 5810/6655



Semi-Supervised Learning

- Access to both labeled and unlabeled samples
- **Goal**: leverage unlabeled data to improve the performance of a method that uses only labeled data
- Requires some kind of smoothness assumption
  - The marginal distribution $p(x)$ contains information about the posterior $p(y|x)$



(a) Smoothness and low-density assumptions. **(b)** Manifold assumption.

Gaussian fields and harmonic functions

- Let $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_m, y_m), \boldsymbol{x}_{m+1}, \ldots, \boldsymbol{x}_{m+n}$ denote the training data
- Let $L = \{1, \ldots, m\}$ and $U = \{m+1, \ldots, m+n\}$
- One of the first (Zhu et al., 2003):

$$f = \arg\min_f \frac{1}{2} \sum_{i,j} w_{ij} \left(f(\boldsymbol{x}_i) - f(\boldsymbol{x}_j) \right)^2$$
$$s.t.\ f(\boldsymbol{x}_i) = y_i, \forall i \in L$$

  - w_{ij} is some measure of similarity (e.g. Gaussian kernel)
- Let $\mathcal{L}$ be the graph Laplacian of the adjacency matrix W
- The solution satisfies $\mathcal{L} f = 0$ (f is harmonic) for unlabeled data

Gaussian fields and harmonic functions

$$f = \arg\min_f \frac{1}{2}\sum_{i,j} w_{ij}\left(f(\boldsymbol{x}_i) - f(\boldsymbol{x}_j)\right)^2$$
$$s.t.\, f(\boldsymbol{x}_i) = y_i, \forall i \in L$$

- The solution satisfies $\mathcal{L}f = 0$ (f is harmonic) for unlabeled data
- Also:

$$f(\boldsymbol{x}_j) = \frac{1}{d_j}\sum_{i\in\mathcal{N}_j} w_{ij} f(\boldsymbol{x}_i),\ j \in U.$$

  - d_j is the degree of $\boldsymbol{x}_j$ and $\mathcal{N}_j$ is its neighborhood
- Matrix notation: $f = D^{-1}Wf = Pf$
- **Solution**: $f_u = (D_{uu} - W_{uu})^{-1}W_{ul}f_l$

Learning with local global consistency

- Zhou et al. (2004):

$$\min_f \frac{1}{2}\sum_{i,j} w_{ij}\left\|\frac{f(x_i)}{\sqrt{d_i}}-\frac{f(x_j)}{\sqrt{d_j}}\right\|^2+\mu\sum_{i\in L}\|f(x_i)-y_i\|^2$$

- Known labels become a soft constraint
- More robust to label noise
- Influence of high degree nodes is normalized
- Solution:

$$f=\beta\left(I-\alpha D^{-\frac{1}{2}}WD^{-\frac{1}{2}}\right)^{-1}Y$$

- Can also be solved iteratively

Linear neighborhood propagation

- Inspired by Locally Linear Embedding
- Learn the weights (Wang et al., 2007):

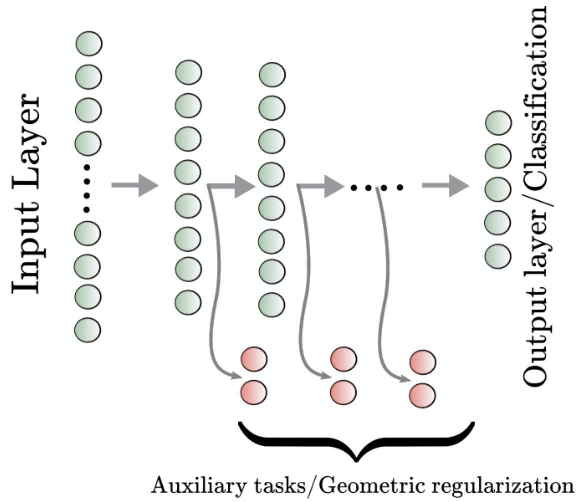
$$W = \arg\min_{W} \left\| x_i - \sum_{j \in \mathcal{N}_i} w_{ij} x_j \right\|^2$$

$$s.t. w_{ij} \geq 0, \sum_{j \in \mathcal{N}_i} w_{ij} = 1$$

- One extension uses a Gaussian kernel and the bandwidth is optimized (Karasuyama et al., 2017)

Semi-supervised learning with GRAE

- We can use GRAE to do semi-supervised learning



- Compared to other graph-based methods

Dataset	Number of classes	Observations	Features
HumanPancreas	14	8569	1000
PBMC	6	1919	50
MNIST	10	5000	784
Xin	4	1449	1000
Zheng Sorted	10	20000	1000
Zheng68k	11	10000	1000

Semi-supervised learning with GRAE

Dataset	Model	Labeled percentage				
		1.00%	5.00%	10.00%	20.00%	40.00%
Human Pancreas	GRNN (Ours)	**0.909** (1)	**0.934** (1)	**0.936** (1)	**0.940** (1)	**0.942** (2)
	Vanilla NN	0.495 (5)	0.693 (5)	0.707 (5)	0.866 (4)	**0.943** (1)
	Laplace	0.765 (3)	0.818 (3)	0.854 (3)	0.884 (3)	0.909 (4)
	p-Laplace	**0.787** (2)	**0.837** (2)	**0.868** (2)	**0.892** (2)	0.913 (3)
	Poisson	0.693 (4)	0.765 (4)	0.796 (4)	0.821 (5)	0.838 (5)
PBMC	GRNN (Ours)	**0.867** (1)	**0.919** (1)	**0.922** (1)	**0.936** (1)	**0.941** (1)
	Vanilla NN	0.389 (5)	0.650 (5)	0.775 (5)	0.865 (5)	0.923 (4)
	Laplace	0.514 (4)	0.809 (4)	0.908 (3)	**0.930** (2)	**0.940** (2)
	p-Laplace	0.628 (3)	0.880 (3)	**0.911** (2)	0.925 (3)	0.934 (3)
	Poisson	**0.857** (2)	**0.899** (2)	0.905 (4)	0.911 (4)	0.919 (5)
Mnist	GRNN (Ours)	**0.884** (1)	**0.929** (1)	**0.947** (1)	0.954 (3)	0.955 (3)
	Vanilla NN	0.563 (5)	0.750 (5)	0.836 (5)	0.894 (5)	0.930 (4)
	Laplace	0.721 (4)	0.917 (3)	**0.943** (2)	**0.957** (1)	**0.966** (1)
	p-Laplace	**0.776** (2)	**0.919** (2)	0.942 (3)	**0.955** (2)	**0.963** (2)
	Poisson	0.775 (3)	0.886 (4)	0.891 (4)	0.902 (4)	0.903 (5)
Xin	GRNN (Ours)	**0.721** (2)	0.831 (3)	0.856 (3)	0.869 (4)	0.885 (3)
	Vanilla NN	0.459 (5)	0.562 (5)	0.590 (5)	0.565 (5)	0.470 (5)
	Laplace	0.645 (3)	**0.857** (2)	**0.876** (2)	**0.896** (2)	**0.903** (2)
	p-Laplace	**0.764** (1)	**0.872** (1)	**0.882** (1)	**0.900** (1)	**0.905** (1)
	Poisson	0.624 (4)	0.816 (4)	0.853 (4)	0.871 (3)	0.885 (3)
Zheng	GRNN (Ours)	**0.732** (1)	**0.784** (1)	**0.807** (1)	**0.812** (2)	**0.823** (2)
	Vanilla NN	0.528 (3)	**0.720** (2)	**0.783** (2)	**0.824** (1)	**0.855** (1)
	Laplace	0.446 (5)	0.545 (5)	0.604 (5)	0.656 (5)	0.691 (4)
	p-Laplace	0.498 (4)	0.565 (4)	0.617 (4)	0.660 (4)	0.691 (4)
	Poisson	**0.635** (2)	0.663 (3)	0.675 (3)	0.688 (3)	0.700 (3)
Zheng 68k	GRNN (Ours)	**0.555** (1)	**0.596** (1)	**0.604** (1)	**0.608** (1)	**0.618** (1)
	Vanilla NN	0.337 (4)	0.455 (4)	0.506 (4)	0.549 (4)	**0.590** (2)
	Laplace	0.390 (3)	0.541 (3)	**0.573** (2)	**0.588** (2)	0.576 (3)
	p-Laplace	**0.460** (2)	**0.549** (2)	0.568 (3)	0.581 (3)	0.572 (4)
	Poisson	0.334 (5)	0.380 (5)	0.390 (5)	0.411 (5)	0.443 (5)

Average testing accuracy is given over 10 runs for various levels of available labeled data. Comparisons are between our approach (GRNN), a Vanilla Neural Network, and three graph-based methods for semi-supervised learning.

Further reading

- Michigan lecture notes: http://web.eecs.umich.edu/~cscott/past_courses/eecs545f16/31_rkhs.pdf
- ESL Sections 5.8 and 12.3.3