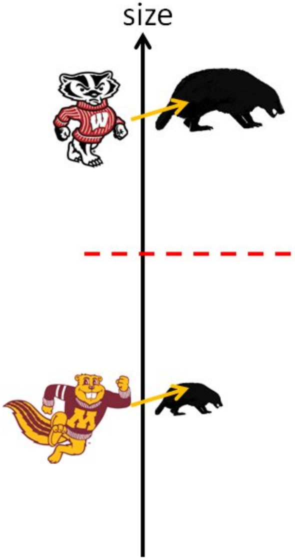
Semi-Supervised Learning by Multi-Manifold Separation

Xiaojin (Jerry) Zhu

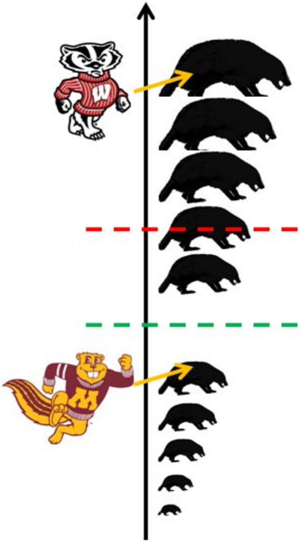
Department of Computer Sciences
University of Wisconsin–Madison

Joint work with Andrew Goldberg, Zhiting Xu, Aarti Singh, and Rob Nowak

Supervised Learning



Semi-Supervised Learning



Prediction Problems

- The feature space $\mathcal{X} = \mathbb{R}^d$
- The label space $\mathcal{Y} = \{0, 1\}$ or $\mathbb{R}$
- Samples $(X, Y) \in \mathcal{X} \times \mathcal{Y} \sim P_{XY}$
  - X: feature vector
  - Y: label
- Goal: construct a *predictor* $f : \mathcal{X} \mapsto \mathcal{Y}$ to minimize

$$R(f) \equiv \mathbb{E}_{(X,Y)\sim P_{XY}}[\mathrm{loss}(Y, f(X))]$$

Learning from Data

- The optimal predictor

$$f^* = \mathrm{argmin}_f \mathbb{E}_{(X,Y)\sim P_{XY}}[\mathrm{loss}(Y, f(X))]$$

depends on P_{XY}, which is often unknown.

- However, we can *learn* a good predictor from a *training set*

$$\{(X_i, Y_i)\}_{i=1}^n \overset{iid}{\sim} P_{XY}$$

- *Supervised Learning*:

$$\{(X_i, Y_i)\}_{i=1}^n \Rightarrow \hat{f}_n$$

Semi-Supervised Learning

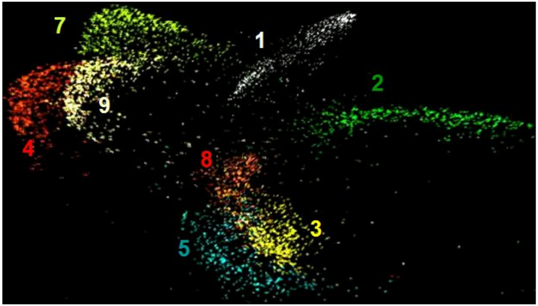
- In many applications in science and engineering, labeled data are scarce, but unlabeled data are abundant and cheap.

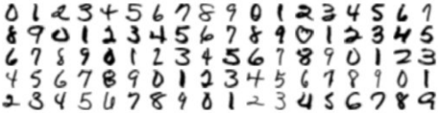
$$\{(X_i, Y_i)\}_{i=1}^n \overset{iid}{\sim} P_{XY}, \{X_j\}_{j=1}^m \overset{iid}{\sim} P_X, m \gg n$$

- *Semi-Supervised Learning (SSL)*:

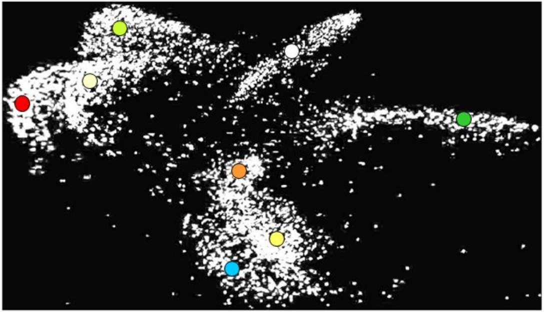
$$\{(X_i, Y_i)\}_{i=1}^n, \{X_j\}_{j=1}^m \Rightarrow \hat{f}_{m,n}$$

Example: Handwritten Digits Recognition





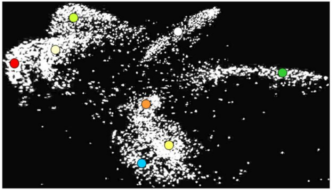
Example: Handwritten Digits Recognition



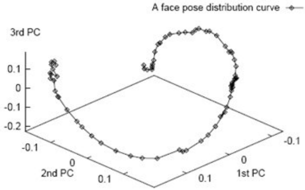
- many unlabeled data + a few labeled data
- knowledge of manifold/cluster + a few labels in each manifold/cluster is sufficient to design a good predictor

Common Assumptions in SSL

- *Cluster assumption*: f^* is constant or smooth on connected high density regions.



- *Manifold assumption*: Support set of P_X lies on low-dimensional manifolds. f^* is smooth wrt geodesic distance on manifolds.



Mathematical Formalization

- Generic Learning Classes:

$$\mathcal{P}_{XY} = \left\{ P_X P_{Y|X} : P_X \in \mathcal{P}_X, P_{Y|X} \in \mathcal{P}_{Y|X} \right\}$$

- "Linked" Learning Classes:

$$\mathcal{P}'_{XY} = \left\{ P_X P_{Y|X} : P_X \in \mathcal{P}_X, P_{Y|X} \in \mathcal{P}_{Y|X}(P_X) \subset \mathcal{P}_{Y|X} \right\}$$

  Link: unlabeled data may inform design of predictor

- SSL can yield faster rate of error convergence than supervised learning:

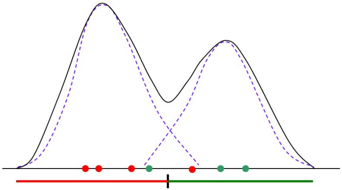
$$\sup_{\mathcal{P}'_{XY}} \mathbb{E}[R(\hat{f}_{m,n})] \leq \inf_{f_n} \sup_{\mathcal{P}'_{XY}} \mathbb{E}[R(f_n)]$$

  - $\hat{f}_n$: predictor based on n labeled examples
  - $\hat{f}_{m,n}$: based on n labeled and m unlabeled examples

The Value of Unlabeled Data

- Castelli and Cover'95 (classification): assume identifiable mixture

$$p(x) = p(x|Y=0)p(Y=0) + p(x|Y=1)p(Y=1)$$



- *Learn* decision regions from (the many) unlabeled examples
- *Label* decision regions from (the few) labeled examples
- Main result:

$$\sup_{\mathcal{P}'_{XY}} \mathbb{E}[R(\hat{f}_{\infty,n})] - R^* \leq Ce^{-\alpha n}$$

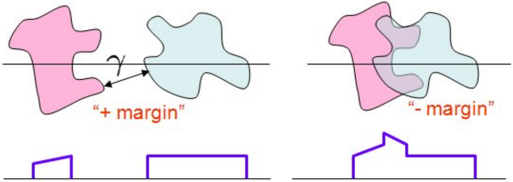
- What about more general cluster or manifold assumptions?

Do Unlabeled Data Help in General?

- No. Lafferty & Wasserman (2007)
  - fix complexity of P_{XY}, let n grow
  - given enough labeled data, unlabeled data is superfluous (no faster rates of convergence for SSL).
- Yes. Niyogi (2008)
  - let complexity of P_{XY} grow with n
  - given finite n, the complexity of the learning problem can be such that supervised learning fails, while SSL has small expected error.
- Both are correct, capturing two extremes. Finite sample bounds give a more complete picture.

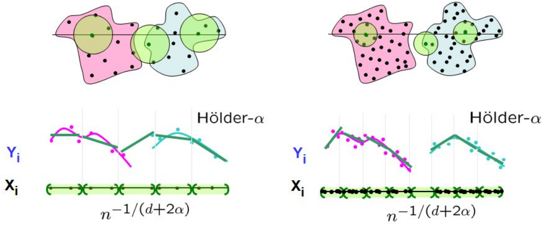
Decision Regions

- Our assumption: $\mathcal{P}'_{XY} \equiv (p_X, f)$
- $\mathrm{supp}(p_X) = \cup_i C_i$, union of compact sets with γ separation



- marginal density p_X bounded away from zero, smooth in C_i
- regression function $f(x)$ Hölder-α smooth on each support set

Adaptive Supervised Learning



- If $\gamma \geq \gamma_0 > 0$ and $n \to \infty$, SL will "discover" decision regions, *eventually* the excess risk of SL (squared loss) is minimax (SSL has no advantage):

$$\sup_{\mathcal{P}'_{XY}} \mathbb{E}[R(\hat{f}_n)] - R^* \leq C n^{-\frac{2\alpha}{2\alpha+d}}$$

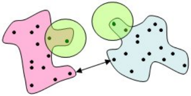
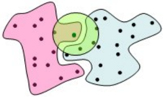
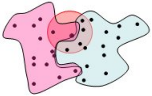
- But, if n fixed and $\gamma \to 0$, *eventually* SL will mix up decision regions and mess up:

$$c n^{-\frac{1}{d}} \leq \inf_{f_n} \sup_{\mathcal{P}'_{XY}} \mathbb{E}[R(f_n)] - R^*$$

- Unlabeled data identify decision regions, mess up later (smaller γ)

Unlabeled Data: Now it helps, now it doesn't

(Singh, Nowak & Zhu, NIPS 2008)

	margin	SSL upper	SL lower	SSL helps?
	$n^{-\frac{1}{d}} \leq \gamma$	$n^{-\frac{2\alpha}{2\alpha+d}}$	$n^{-\frac{2\alpha}{2\alpha+d}}$	no
	$m^{-\frac{1}{d}} \leq \gamma < n^{-\frac{1}{d}}$	$n^{-\frac{2\alpha}{2\alpha+d}}$	$n^{-\frac{1}{d}}$	yes
	$-m^{-\frac{1}{d}} \leq \gamma < m^{-\frac{1}{d}}$	$n^{-\frac{1}{d}}$	$n^{-\frac{1}{d}}$	no
	$\gamma < -m^{-\frac{1}{d}}$	$n^{-\frac{2\alpha}{2\alpha+d}}$	$n^{-\frac{1}{d}}$	yes

An SSL Algorithm

Given n labeled examples and m unlabeled examples,

1. Use unlabeled data to infer $\log(n)$ decision regions $\hat{C}_i$
   - plug in your favorite manifold clustering algorithm that detects abrupt change in support, density, dimensionality, etc.
   - carve up ambient space into $\hat{C}_i$: Voronoi
   - each $\hat{C}_i$ has to be "big enough", $\geq n/\log^2(n)$ labeled examples, $\geq m/\log^2(n)$ unlabeled examples
2. Use the labeled data in $\hat{C}_i$ to train SL $\hat{f}_i$
3. If a test point $x^* \in \hat{C}_i$, predict $\hat{f}_i(x^*)$

Similar to "cluster & label".

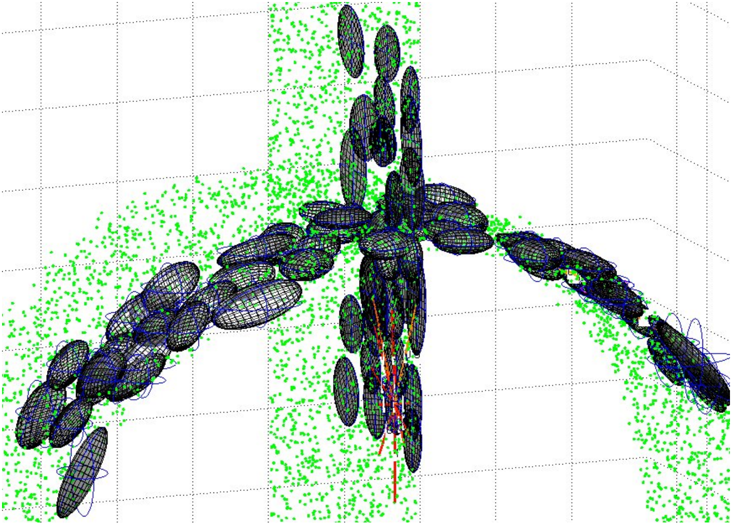
Building Blocks: Local Covariance Matrix

- For $x \in \{x_i\}_{i=1}^{n+m}$, find its $[\log(n+m)]$ nearest neighbors (in Euclidean distance)
- The local covariance matrix of the neighbors

$$\Sigma_x = \frac{1}{[\log(n+m)] - 1} \sum_j (x_j - \mu_x)(x_j - \mu_x)^\top$$

- Σ_x captures local geometry

Building Blocks: Local Covariance Matrix



A Distance Between Σ_1 and Σ_2

- Hellinger distance

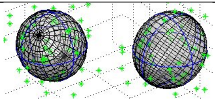
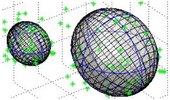
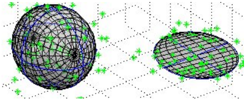
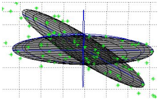
$$H^2(p,q) = \frac{1}{2}\int \left(\sqrt{p(x)} - \sqrt{q(x)}\right)^2 dx$$

- $H(p,q)$ symmetric, in $[0,1]$
- Let $p = N(0,\Sigma_1), q = N(0,\Sigma_2)$. We define

$$H(\Sigma_1,\Sigma_2) = \sqrt{1 - 2^{\frac{d}{2}} \frac{|\Sigma_1|^{\frac{1}{4}}|\Sigma_2|^{\frac{1}{4}}}{|\Sigma_1+\Sigma_2|^{\frac{1}{2}}}}$$

(computed in common subspace)

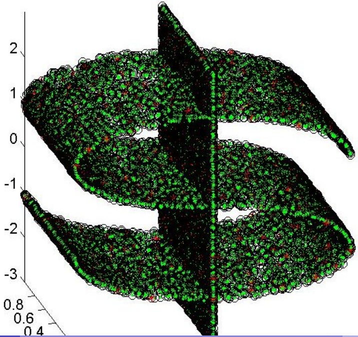
Hellinger Distance

	Comment	$H(\Sigma_1, \Sigma_2)$
	similar	0.02
	density	0.28
	dimension	1
	orientation*	1

* smoothed version: $\Sigma + \epsilon I$

A Sparse Subset

- Two close points will have similar neighbors $\Rightarrow$ small H even they are on different manifolds
- Compute a sparse subset of $m' = \frac{m}{\log(m)}$ points (red dots):
  - Start from an arbitrary x^0
  - Remove its $\log(m)$ nearest neighbors
  - Let x^1 be the next nearest neighbor, repeat
- Include all labeled data
- Random sampling might work too



A Sparse Graph on the Sparse Subset

- Sparse nearest neighbor graph on the sparse subset, use Mahalanobis distance to trace the manifold

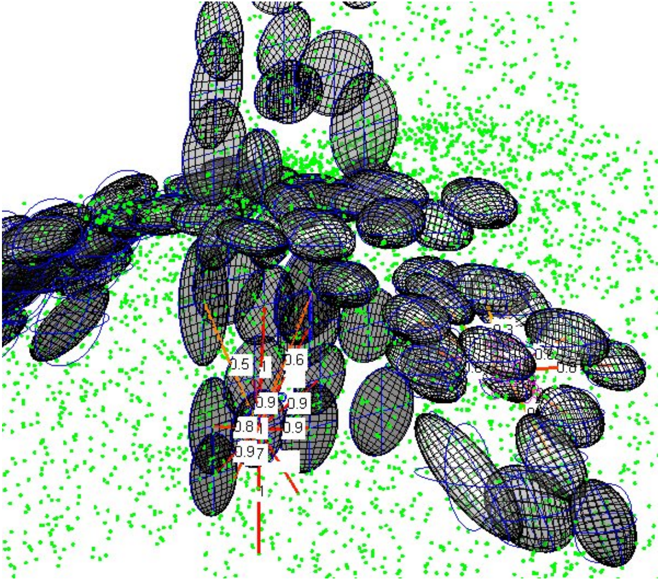
$$d^2(x, y) = (x - y)^\top \Sigma_x^{-1} (x - y)$$

- Gaussian edge weight on sparse edges

$$w_{ij} = e^{-\frac{H^2(\Sigma_{x_i}, \Sigma_{x_j})}{2\sigma^2}}$$

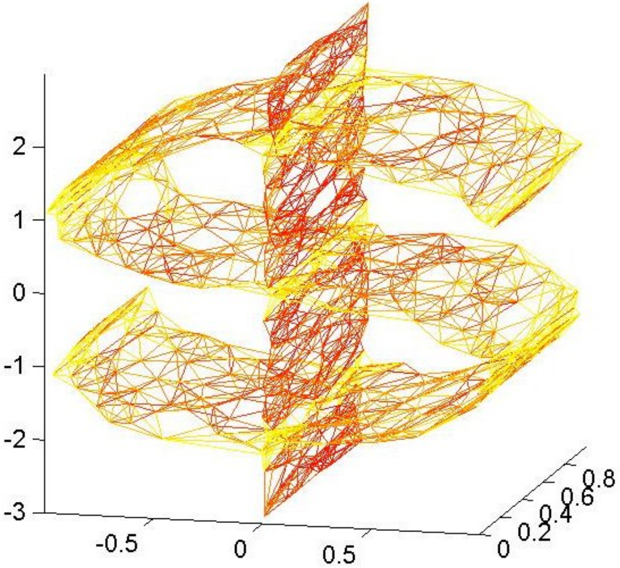
- Combines locality and shape

A Sparse Graph on the Sparse Subset



A Sparse Graph on the Sparse Subset

Red=large w, yellow=small w



Multi-Manifold Separation as Graph Cut

- Cut the graph $W = [w_{ij}]$ into $k \equiv [\log(n)]$ parts
- Each part has at least $n/\log^2(n)$ labeled examples, $m'/\log^2(n)$ unlabeled examples
- Formally: RatioCut with size constraints
  - Let the k parts be $A_1, \ldots, A_k$
  - cut$(A_i, \bar{A}_i) = \sum_{s \in A_i, t \in \bar{A}_i} w_{st}$
  - cut$(A_1, \ldots, A_k) = \sum_{i=1}^{k} cut(A_i, \bar{A}_i)$
  - Minimize cut directly tend to produce very unbalanced parts
  - RatioCut$(A_1, \ldots, A_k) = \sum_{i=1}^{k} \frac{cut(A_i, \bar{A}_i)}{|A_i|}$
  - However, this balancing heuristic may not satisfy our size constraints

RatioCut Approximated by Spectral Clustering

Well-known RatioCut approximation (without size constraints) [e.g., von Luxburg 2006]

- Define k indicator vectors $h_1, \ldots, h_k$

$$h_{ij} = \begin{cases} 1/\sqrt{A_j} & \text{if } i \in A_j \\ 0 & \text{otherwise} \end{cases}$$

- Matrix H has columns $h_1, \ldots, h_k$, $H^\top H = I$
- RatioCut$(A_1, \ldots, A_k) = \frac{1}{2} tr(H^\top L H)$
- $\min_H tr(H^\top L H)$ subject to $H^\top H = I$
- Relax elements of H to $\mathbb{R} \Rightarrow$ [Rayleigh-Ritz] $h_1, \ldots, h_k$ are the first k eigenvectors of L.
- "Un-relax" H to hard partition: k-way clustering

RatioCut Approximated by Spectral Clustering

The spectral clustering algorithm (without size constraints):

1. Unnormalized Laplacian $L = D - W$
2. First k eigenvectors $v_1, \ldots, v_k$ of L
3. V matrix: $v_1, \ldots, v_k$ as columns, $n + m'$ rows
4. New representation of x_i: the ith row of V
5. Cluster x_i into k clusters with k-means. The clusters define $A_1, \ldots, A_k$.

Next: enforce size constraints in k-means.

Standard (Unconstrained) k-Means

- k-means clusters $x_1 \dots x_N$ into k clusters with center $C_1 \dots C_k$:

$$\min_{C_1 \dots C_k} \sum_{i=1}^{N} \min_{h=1 \dots k} \left(\frac{1}{2} \|x_i - C_h\|^2 \right)$$

- Introduce indicator matrix T, $T_{ih} = 1$ if x_i belongs to C_h

$$\begin{aligned} \min_{C,T} \quad & \textstyle\sum_{i=1}^{N} \sum_{h=1}^{k} T_{ih} \left(\frac{1}{2} \|x_i - C_h\|^2 \right) \\ \text{s.t.} \quad & \textstyle\sum_{h=1}^{k} T_{ih} = 1, T \geq 0 \end{aligned}$$

- Local optimum found by starting from arbitrary $C_1 \dots C_k$ and iterating:
  1. Update $T_{i\cdot}$: assign each point x_i to its closest center C_h
  2. Update centers $C_1 \dots C_k$ by the mean of points assigned to that center

  Note: each step reduces the objective.

Size Constrained k-Means

(Bradley, Bennett, Demiriz. 2000)

- Size constraints: cluster h must have at least τ_h points

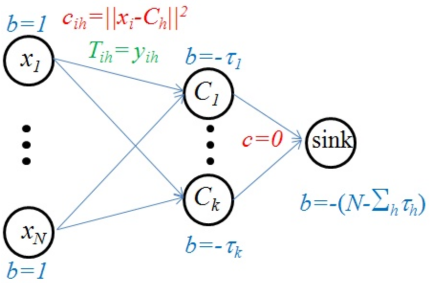
$$\begin{aligned} \min_{C,T} \quad & \textstyle\sum_{i=1}^{N} \sum_{h=1}^{k} T_{ih} \left(\frac{1}{2} \|x_i - C_h\|^2 \right) \\ \text{s.t.} \quad & \textstyle\sum_{h=1}^{k} T_{ih} = 1, T \geq 0 \\ & \textstyle\sum_{i=1}^{N} T_{ih} \geq \tau_h, h = 1 \ldots k. \end{aligned}$$

- Solving T looks like a difficult integer problem
- Surprise: efficient integer solution found by Minimum Cost Flow linear program

Minimum Cost Flow

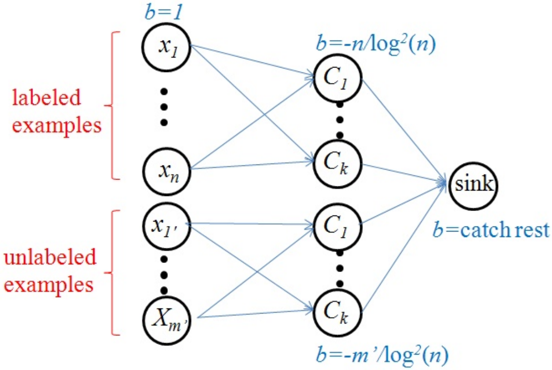
- A graph with supply nodes ($b_i > 0$) and demand nodes ($b_i < 0$); directed edge $i \to j$ has unit transportation cost c_{ij}, traffic variable $y_{ij} \geq 0$; Meet demand with minimum transportation cost

$$\begin{aligned} \min_{y} \quad & \textstyle\sum_{i \to j} c_{ij} y_{ij} \\ \text{s.t.} \quad & \textstyle\sum_j y_{ij} - \sum_j y_{ji} = b_i \end{aligned}$$

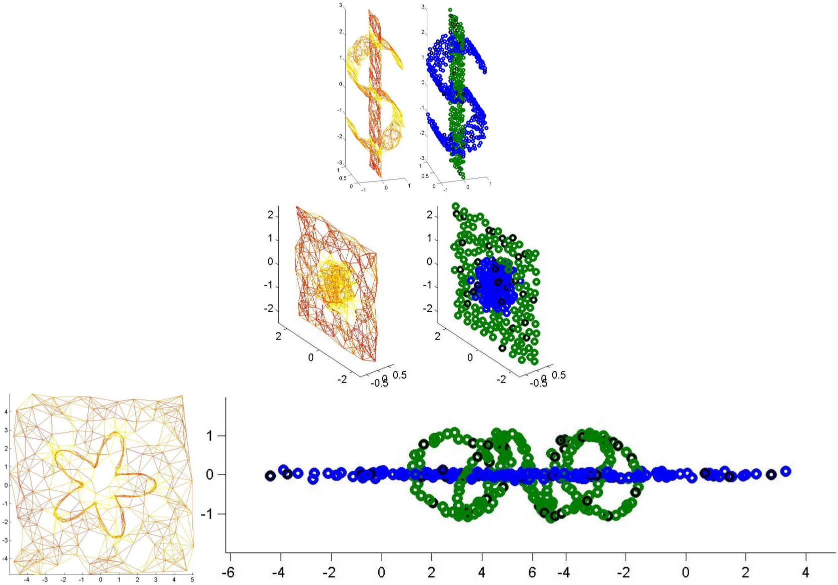


Minimum Cost Flow with Two Constraints

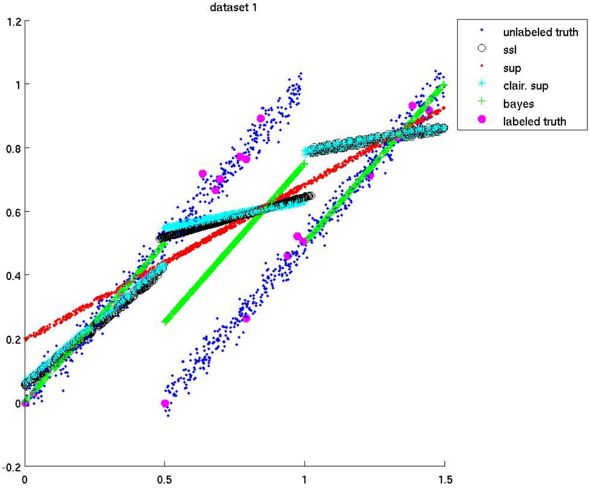
Each cluster has at least $n/\log^2(n)$ labeled examples, $m'/\log^2(n)$ unlabeled examples



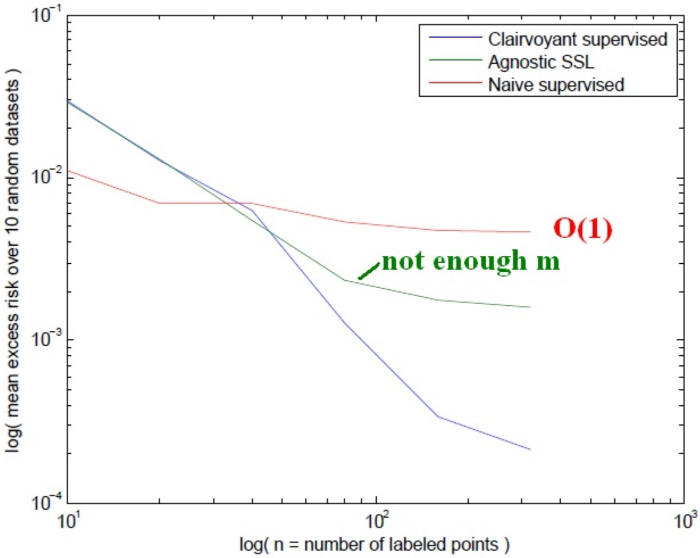
Example Cuts



SSL Example: Two Squares



SSL Example: Two Squares



SSL Example: Dollar Sign