# Semi-Supervised Learning with Label Propagation

**Xiaojin Zhu**
**Zoubin Ghahramani** *
**John Lafferty**

School of Computer Science
Carnegie Mellon University

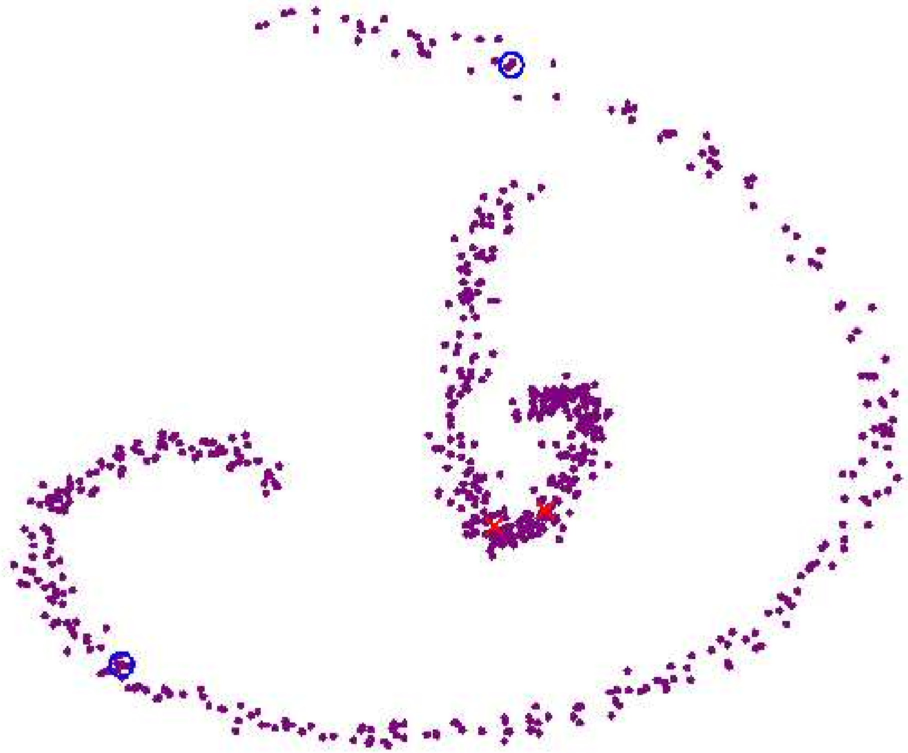
* Gatsby Computational Neuroscience Unit
University College London
{zhuxj, zoubin, lafferty}@cs.cmu.edu

January 13, 2003

# Poverty of the Stimulus

## Classification using Unlabeled Data

Assume: there is information in the data manifold.



# The Problem Statement

Let there be $l$ labeled data $(x_1, y_1) \dots (x_l, y_l)$, $y \in \{0, 1\}$.

Let there be $u$ unlabeled data $x_{l+1} \dots x_{l+u}$; usually $u \gg l$.

We create an **undirected weighted graph**, where the nodes are data points, both labeled and unlabeled.

Assume we are given the **weight matrix** $W$, e.g. $w_{ij} = \exp\left(-\frac{d_{ij}^2}{\sigma^2}\right)$

**Problem**: infer $y$ for unlabeled data (transduction).

We first work out a **real function** $f$ on the graph.

## Label Propagation Algorithm

We want nearby points to have similar labels.

Let $D$ be the diagonal matrix, $D_{ii} = \sum_j w_{ij}$. (the **volume** of node $i$)

Let $P = D^{-1}W$ be the **transition matrix**. ($W$ row normalized)

**Algorithm** Repeat until converge:

1. Propagate labels on all nodes for one step $f = Pf$.

2. Clamp $f(i) = y_i$, for $i = 1 \ldots l$.

## Convergence of $f$

Notation: $W = \begin{bmatrix} W_{ll} & W_{lu} \\ W_{ul} & W_{uu} \end{bmatrix}$, same for $D, P$ etc.

The algorithm computes the stationary point

$$\begin{bmatrix} f_l \\ f_u \end{bmatrix} = \begin{bmatrix} I & 0 \\ P_{ul} & P_{uu} \end{bmatrix} \begin{bmatrix} f_l \\ f_u \end{bmatrix}$$

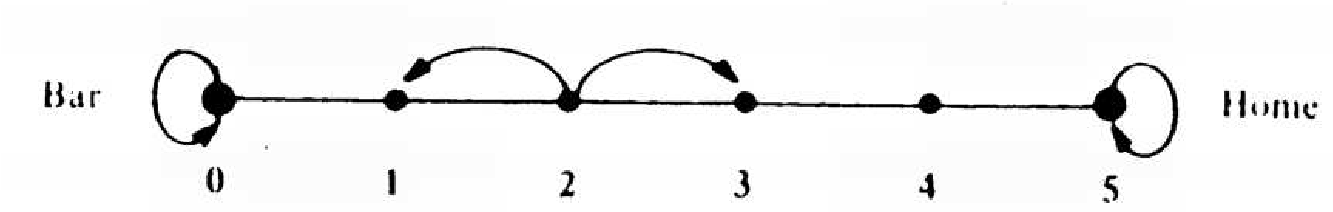
The unique closed form solution is

$$\begin{aligned} f_u &= (I - P_{uu})^{-1} P_{ul} f_l \\ &= (D_{uu} - W_{uu})^{-1} W_{ul} f_l \end{aligned}$$

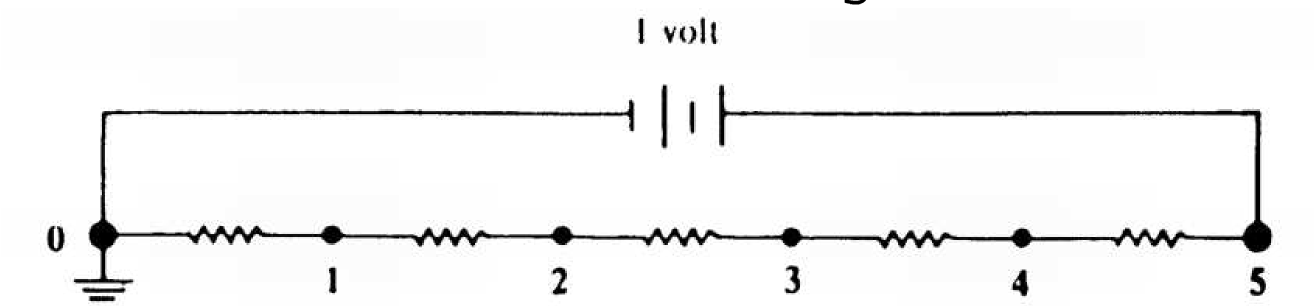
## Random Walks and Electric Networks

(Doyle and Snell, 1984)

**Random walks**: What is the probability that starting from vertex $i$ the random walk will reach "Home" before "Bar"?




**Electric networks**: What is the voltage at vertex $i$?



## Spectral Graph Theory

The **Laplacian** is $L = D - W$. The **heat kernel** is $K_t = e^{-tL}$.

The **Green's function** $G$ is the inverse operator of the restricted Laplacian $L_{uu}$, which is also the integration of heat kernels over time $t$ on unlabeled points:

$$G = \int_0^{\infty} e^{-tL_{uu}} dt = L_{uu}^{-1} = (D_{uu} - W_{uu})^{-1}$$

$f$ is the solution to $Lf = 0$ satisfying the boundary condition $f_l$.

$$f_u = G W_{ul} f_l$$

## Spectral Clustering: Normalized Cut

Both optimize the same **energy function**

$$\sum_{i,j} \frac{1}{2}(f(i) - f(j))^2 w_{ij} = f^{'} L f$$

- Our $f$ is **constrained** on $f_l$;

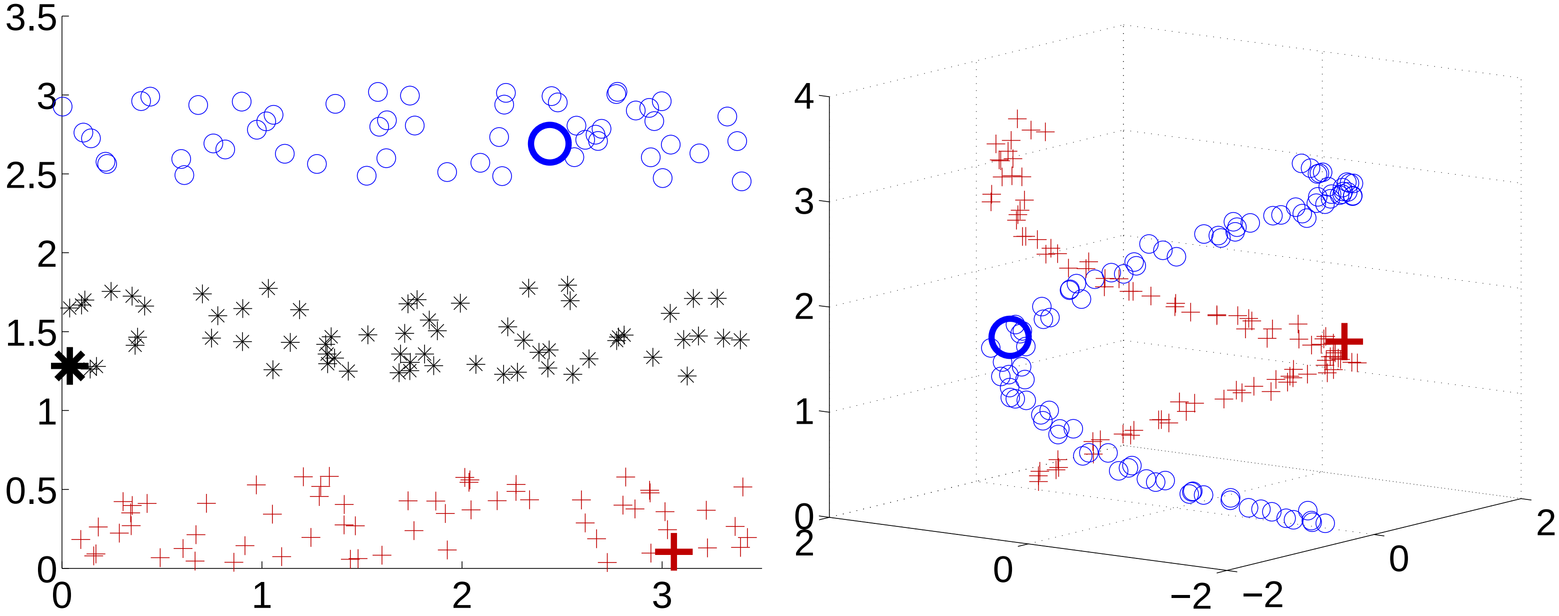
- Normalized cut is not constrained; uses the eigenvector of the second smallest eigenvalue $\lambda_1$ ($\lambda_0 = 0$ has constant eigenvector, useless for segmentation).

"Cluster then label" might be less desirable when classes not well separated.

# From $f$ to Classification

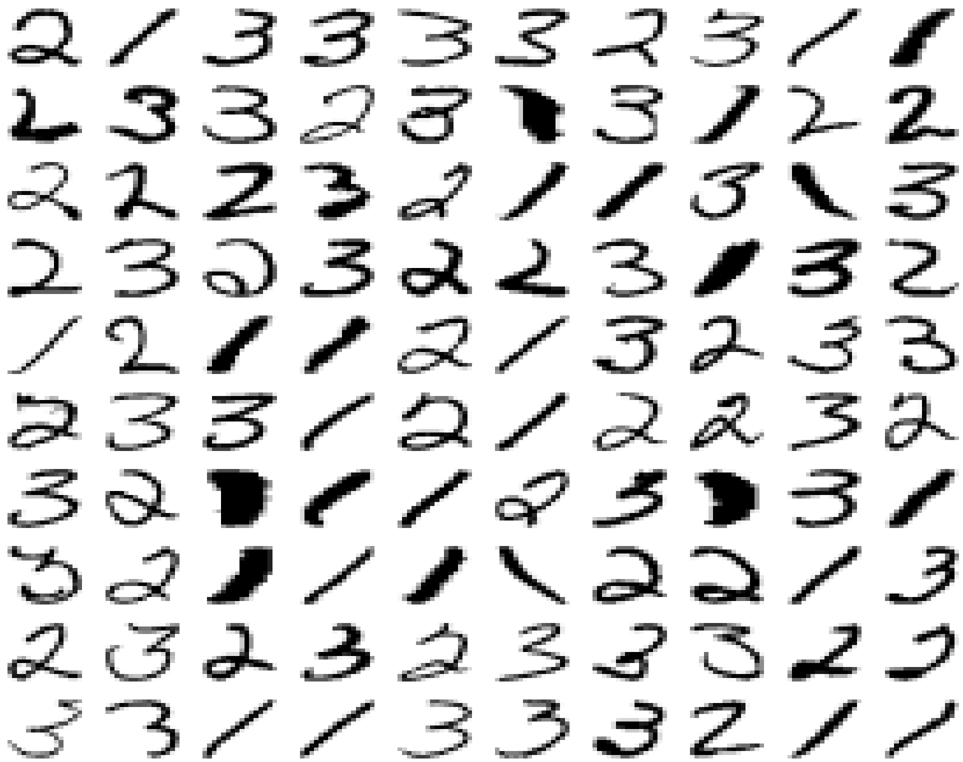
Binary: threshold $f$ at 0.5

Multi-way: one against all

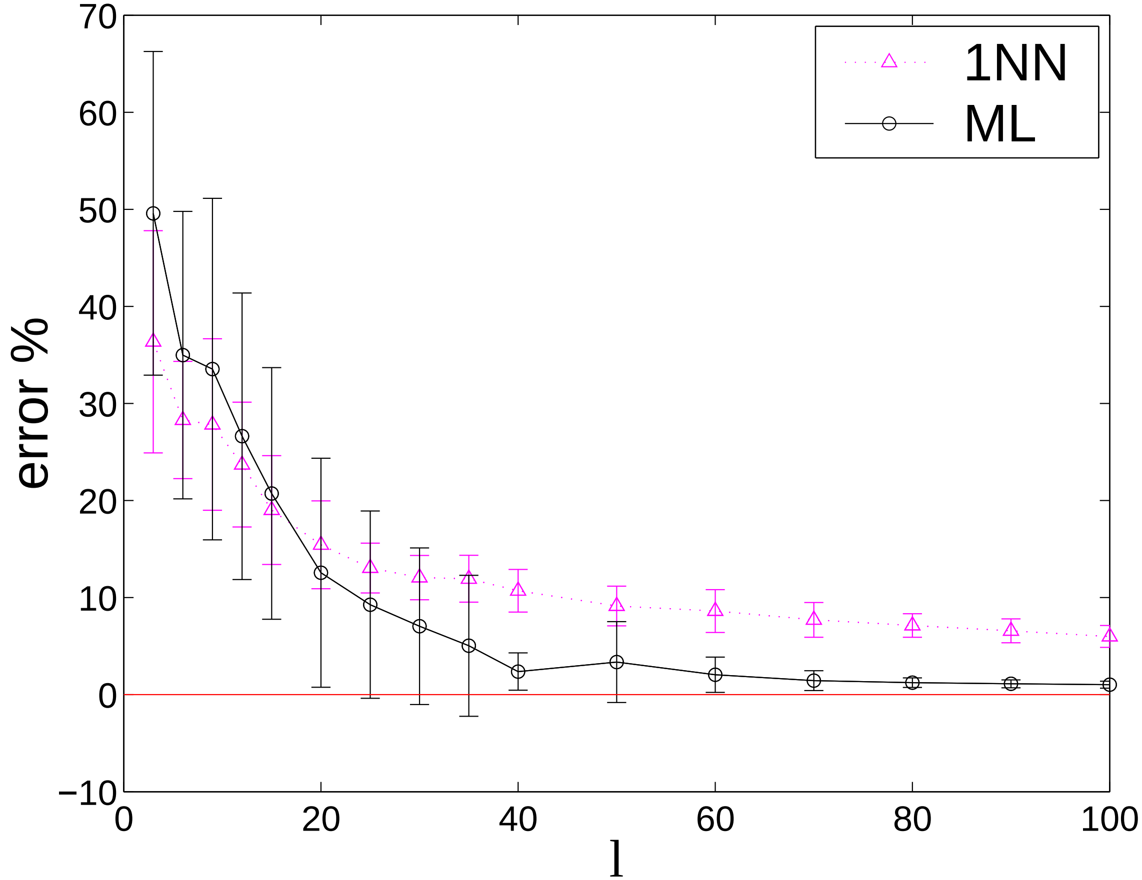


## Handwritten Digits

Cedar Buffalo Digits Dataset $16 \times 16$ grayscale
Digits 1,2,3
1100 images per class.

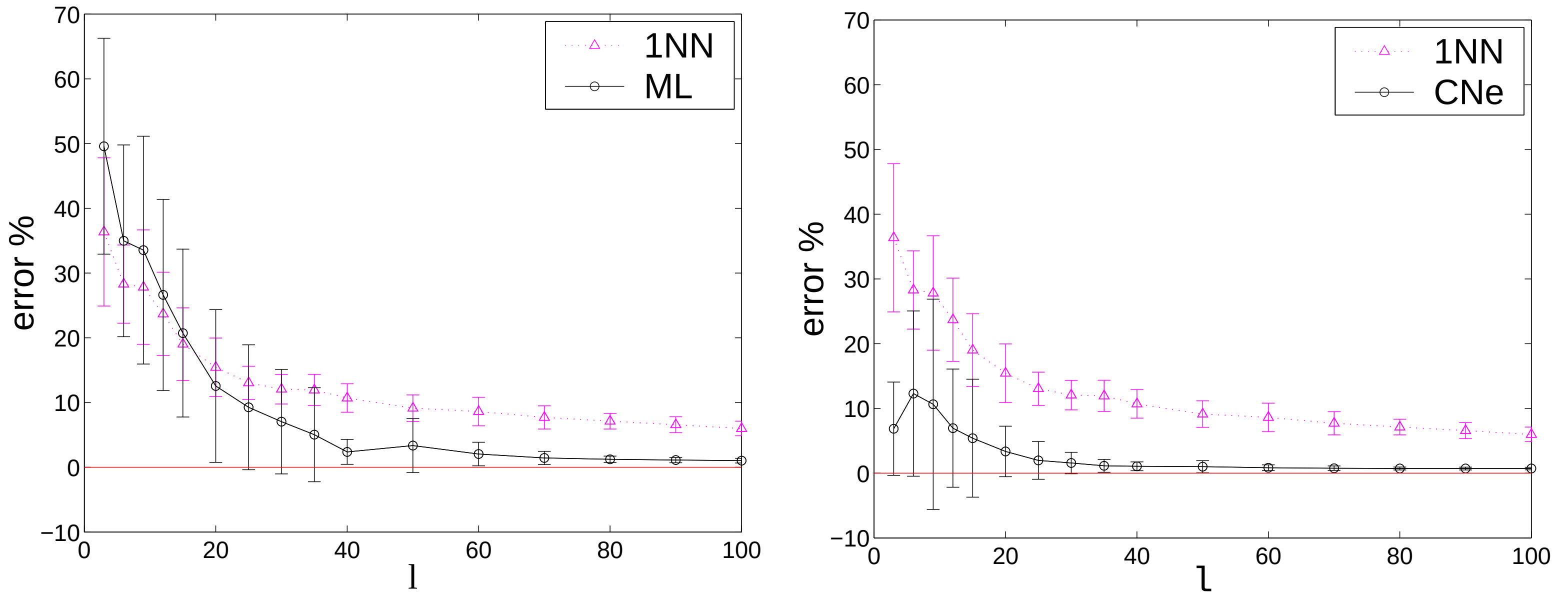


# Results on Handwritten Digits: Un-rebalanced




ML: no rebalancing. Class 1 if $f_i > 1 - f_i$.

## Results on Handwritten Digits: Rebalanced




CNe: Rebalancing. Class 1 if $\frac{q}{\sum f} f_i > \frac{1-q}{\sum(1-f)}(1 - f_i)$.

$q$ is the estimated proportion of class 1 from labeled data.

## Tricky Balance: Maintain Class Proportions

- $\frac{q}{\sum f} f_i > \frac{1-q}{\sum(1-f)}(1 - f_i)$, or

- Constrain $f$ s.t. $\sum f = nq$, or

- Add 2 virtual nodes to graph?

## Learning the weight matrix $W$

Parameterize $W$ with $\sigma_d$, length scale in each dimension.

$$w_{ij} = \exp\left(-\sum_d \frac{(x_i^d - x_j^d)^2}{\sigma_d^2}\right)$$

**Intuition**: we want the most decisive classification of the unlabeled data. Since there are very few hyperparameters, this should not lead to overfitting.

**Minimize Entropy**:

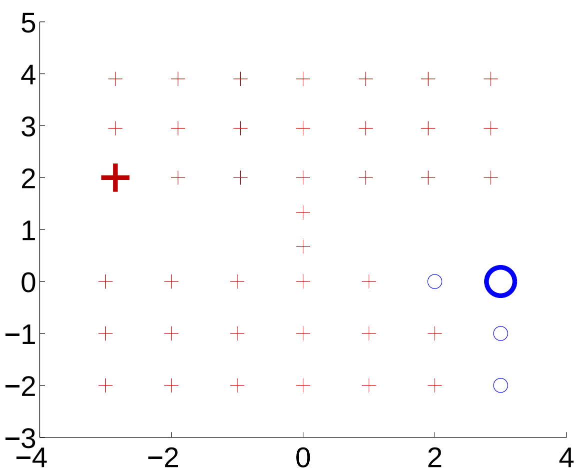
$$H = \sum_{i=l+1}^{l+u} -f_i \log f_i - (1 - f_i) \log(1 - f_i)$$

**Problem**: $\sigma \rightarrow 0$ has $H = 0$ but results in "propagate 1NN" (p1NN) algorithm.

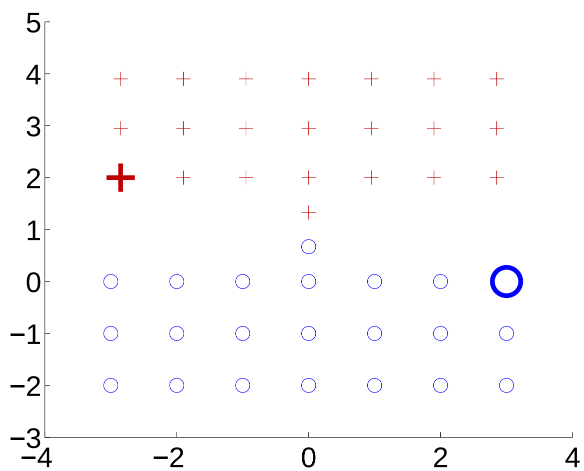
## Learning the weight matrix $W$: Minimum Entropy

**Solution**: smooth $P$ with a uniform transition matrix (like Google's PageRank algorithm...)

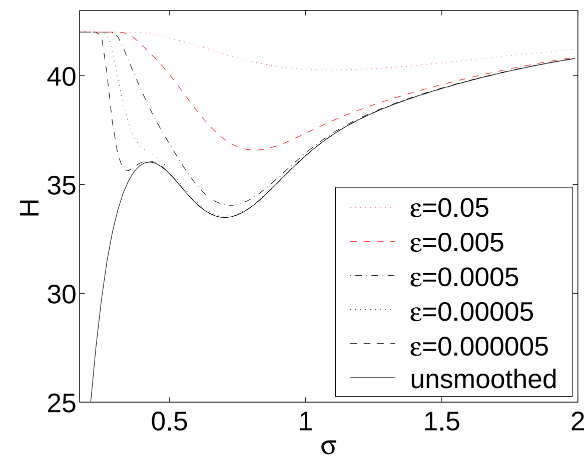
$$\tilde{P} = (1-\epsilon)P + \epsilon U$$



p1NN or $\sigma \to 0$

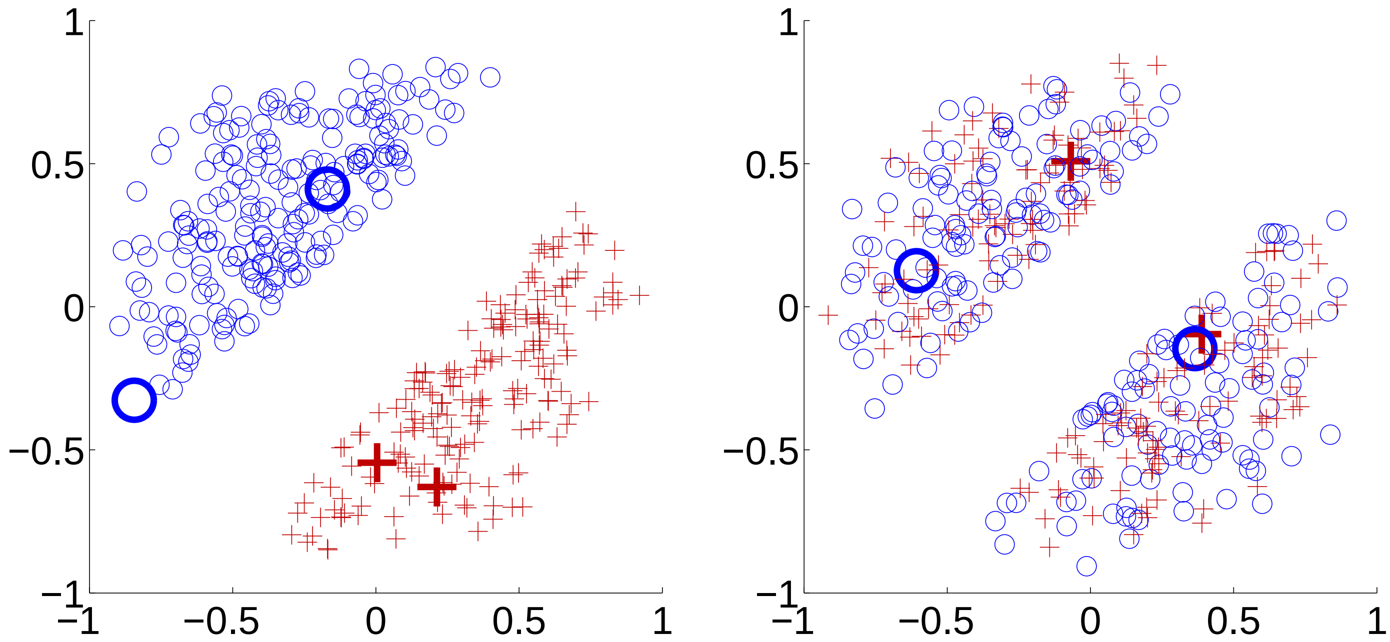


Optimal $\sigma = 0.72$




Smoothing

# Learning the weight matrix $W$




dim 1,2 dim 3,4

$\sigma_1 = 0.18, \sigma_2 = 0.19, \sigma_3 = 14.8, \sigma_4 = 13.3$. With **4 labeled points**, the algorithm learns that dim 1,2 are relevant but dim 3,4 are irrelevant to classification, even though the data are **clustered in dim 3,4** too.

## Summary

- $f$ has many nice properties.

- What is happening in rebalancing?

- Other ways to learn $W$?

**Ref.** Learning from Labeled and Unlabeled Data with Label Propagation. Xiaojin Zhu, Zoubin Ghahramani. CMU CALD tech report CMU-CALD-02-107, 2002.