Semi-Supervised Discriminant Analysis

Dit-Yan Yeung

Department of Computer Science and Engineering
Hong Kong University of Science and Technology

MLA'08

Contents

Related Topics

Semi-Supervised Learning

Discriminant Analysis

Semi-Supervised Learning

Two Most Mature Learning Paradigms: Supervised and Unsupervised Learning

Unsupervised learning:

- Given:
  $X = \{x_1, \ldots, x_n\} \subset \mathcal{X}$, a set of n examples drawn i.i.d. from some (unknown) distribution on the input space $\mathcal{X}$.
- Goal:
  Find interesting structure in X.
- Fundamentally a density estimation problem.
- Weaker forms:
  e.g., quantile estimation, clustering, outlier detection, dimensionality reduction.
- *No supervisory information is available for any training example.*

Two Most Mature Learning Paradigms: Supervised and Unsupervised Learning

Supervised learning:

- Given:
  $X = \{(x_1, y_1), \ldots, (x_n, y_n)\} \subset \mathcal{X} \times \mathcal{Y}$, a set of n i.i.d. pairs where $y_i \in \mathcal{Y}$ is the label or target of input x_i.
- Goal:
  Predict labels or targets of unseen test examples as accurately as possible.
- Most common tasks: classification and regression.
- *Supervisory information is available for all training examples.*

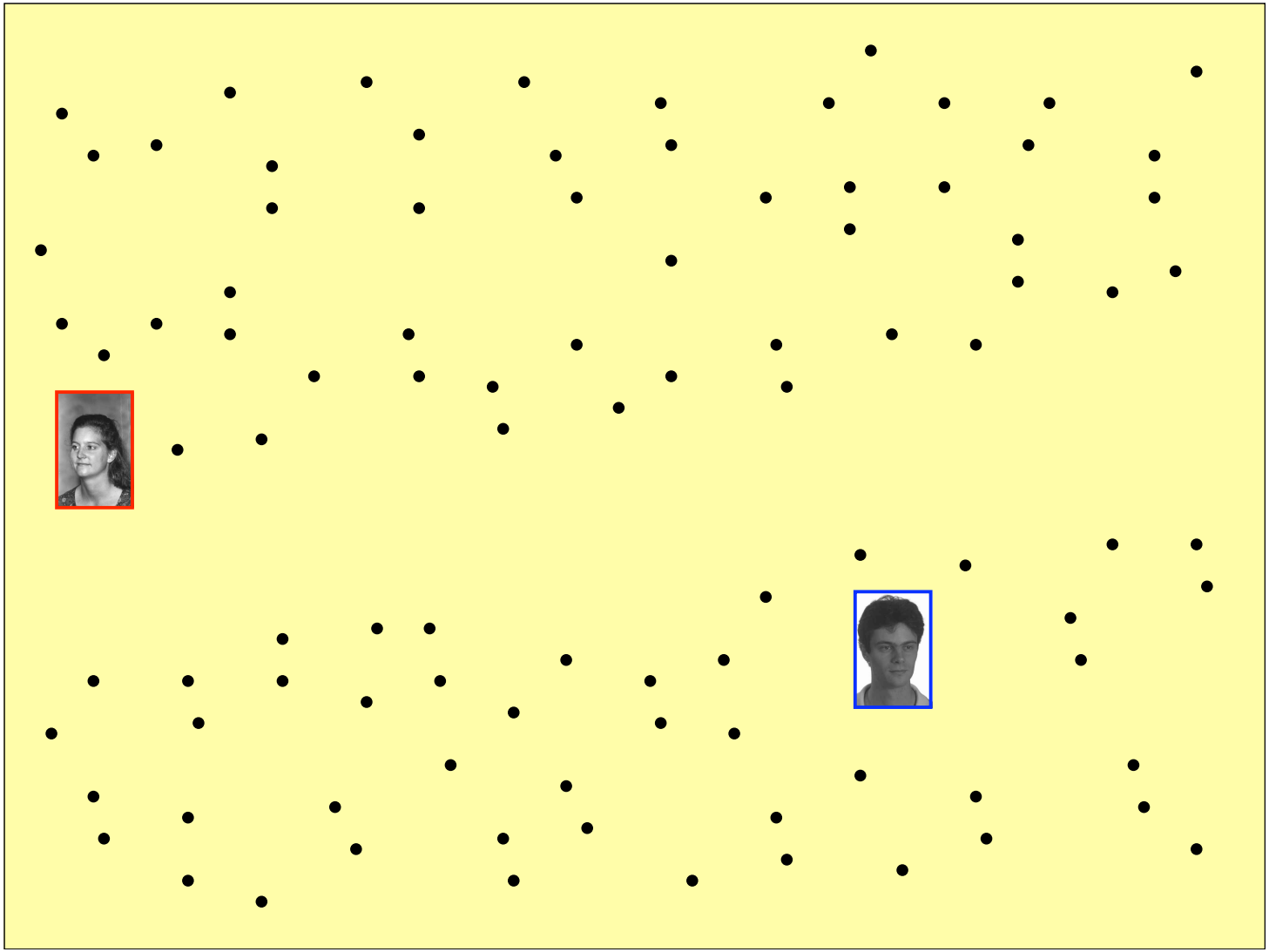
Semi-Supervised Learning

- SSL is halfway between supervised and unsupervised learning.
- *Supervisory information is available for some, but not all, training examples.*

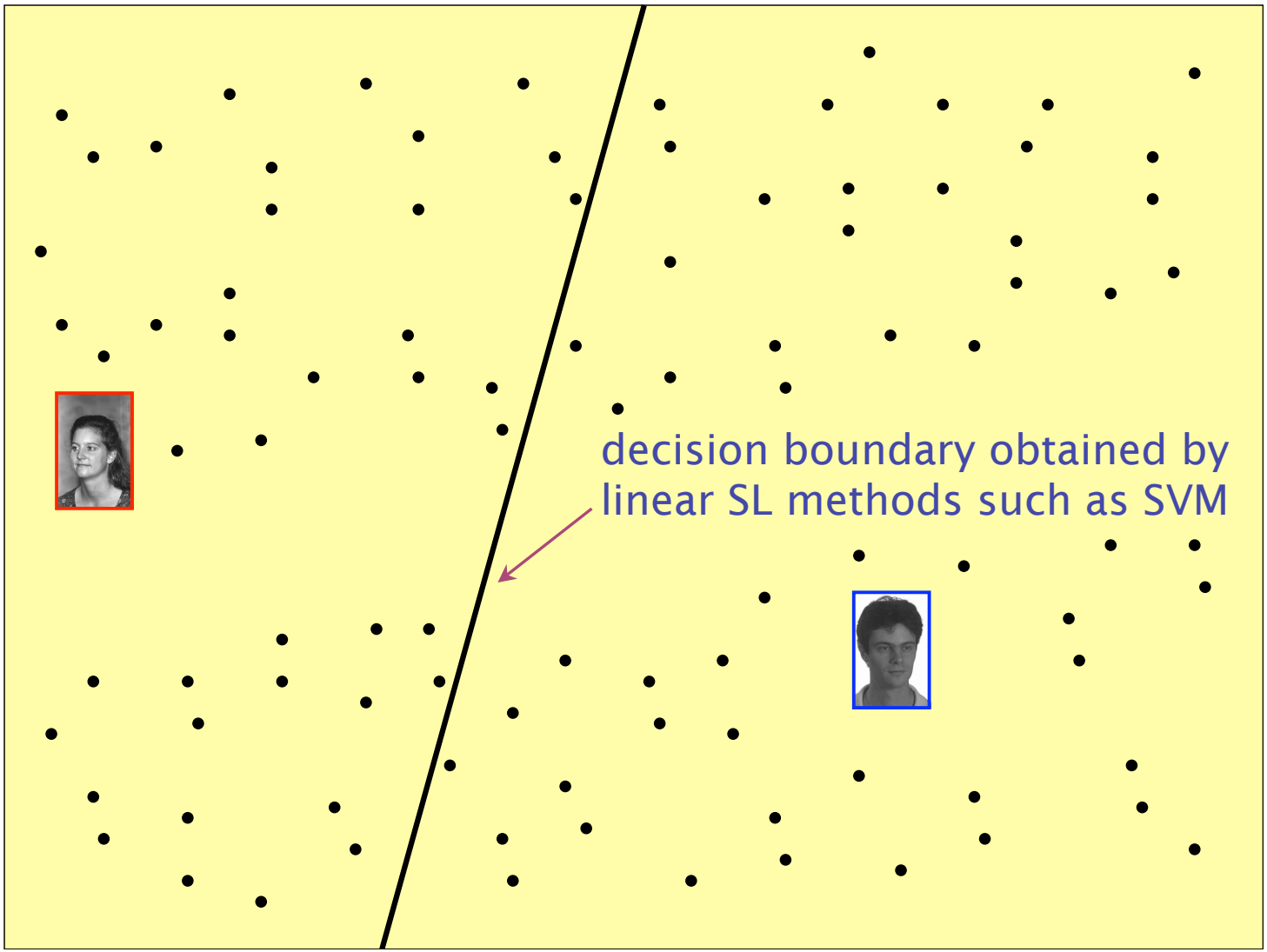
- SSL may be regarded as:
  - Supervised learning augmented with unlabeled data
    e.g., semi-supervised classification, semi-supervised regression
  - Unsupervised learning augmented with labeled data or constraints between data points
    e.g., semi-supervised clustering

- Our focus:
  Semi-supervised classification – most common type of SSL problems studied so far.

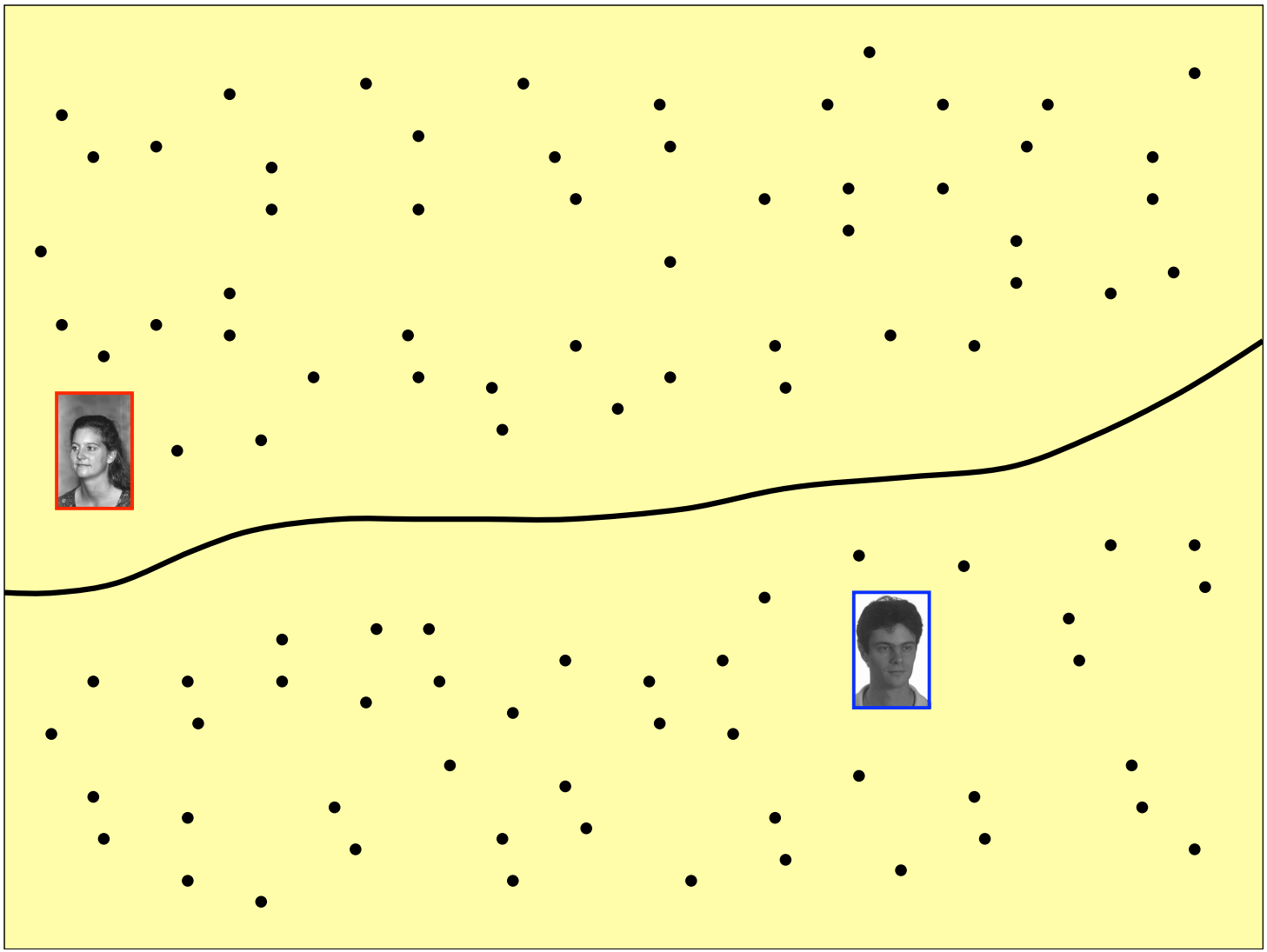
Semi-Supervised Classification Example: Face Recognition



Semi-Supervised Classification Example: Face Recognition

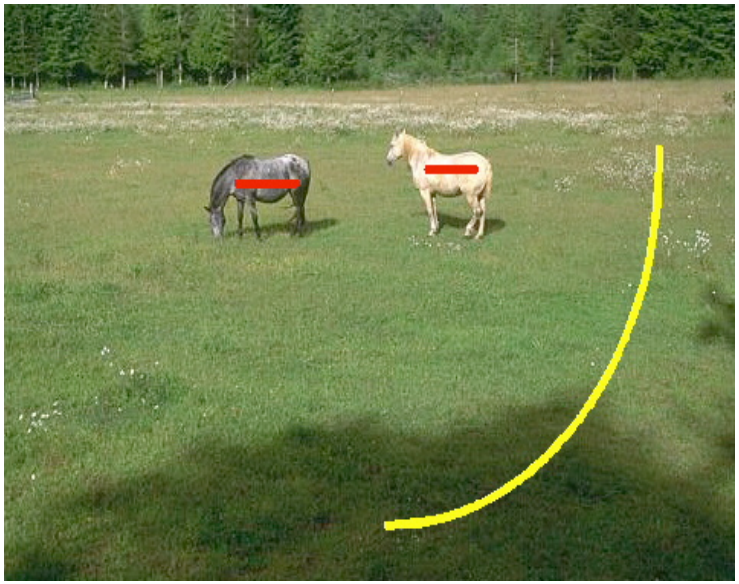


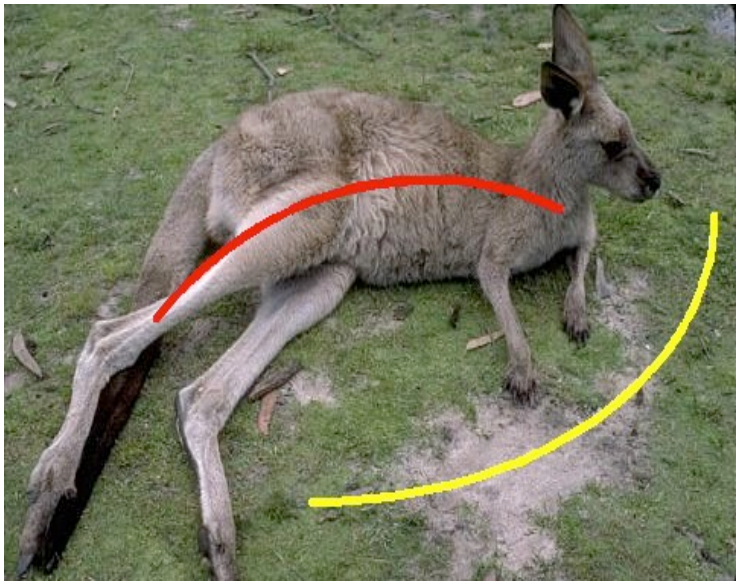
Semi-Supervised Classification Example: Face Recognition



Semi-Supervised Classification Example: Face Recognition

Semi-Supervised Clustering Example: Image Segmentation





When Can SSL Work?

- SSL will yield an improvement over supervised learning if: Knowledge on $p(x)$ gained through unlabeled data carries information that is useful in the inference of $p(y|x)$.
- Failure to meet this requirement may lead to degradation in prediction accuracy by misguiding the inference.

Smoothness Assumption

- For SSL to work, certain assumptions about the data have to hold.
- Semi-supervised smoothness assumption:

  Assumption: If two points x_1, x_2 in a high-density region are close, then so should be the corresponding outputs y_1, y_2.

  Smoothness assumption of supervised learning (for comparison):

  Assumption: If two points x_1, x_2 are close, then so should be the corresponding outputs y_1, y_2.

Cluster Assumption

- Cluster assumption:

  Assumption: If points are in the same cluster, they are likely to be of the same class.

  Equivalent formulation of cluster assumption:
  Low-density separation:

  Assumption: The decision boundary should lie in a low-density region.

- The cluster assumption can be seen as a special case of the semi-supervised smoothness assumption.

Manifold Assumption

- Manifold assumption:

  Assumption: The high-dimensional data lie roughly on a low-dimensional manifold.

- If the data lie on a low-dimensional manifold, then the learning algorithm can essentially operate in a space of corresponding dimensionality, thus avoiding the curse of dimensionality.

Major SSL Models

Generative models:

- Mixture models with missing data.
- Unlabeled data may be used to define data-dependent priors (e.g., over functions).

Low-density separation models:

- E.g., transductive SVM (TSVM) [Joachims, 1999] (with loss function modified to incorporate unlabeled data)

Major SSL Models

Graph-based models:

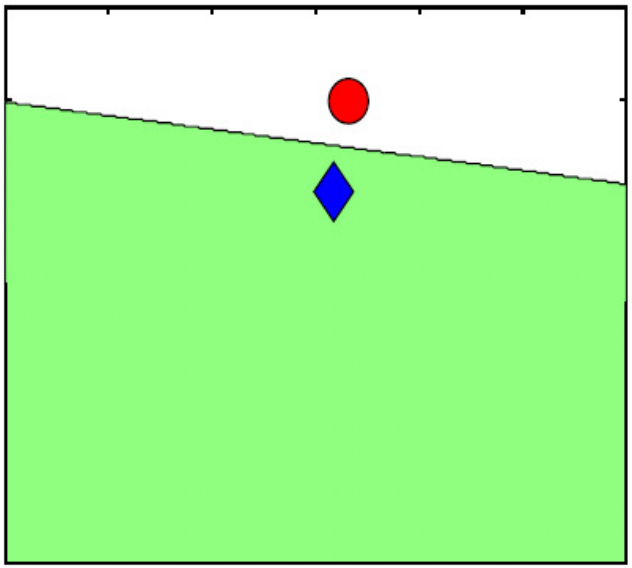
- Most actively studied SSL models.
- Data (both labeled and unlabeled) are represented by nodes of a graph, with edges labeled with pairwise distances between incident nodes.
- Approximate geodesic distance between two points is computed w.r.t. manifold of data points.
- Based on manifold assumption.
- Most existing methods are transductive, with very few exceptions, e.g., Laplacian SVM (LapSVM) [Belkin et al., 2005].

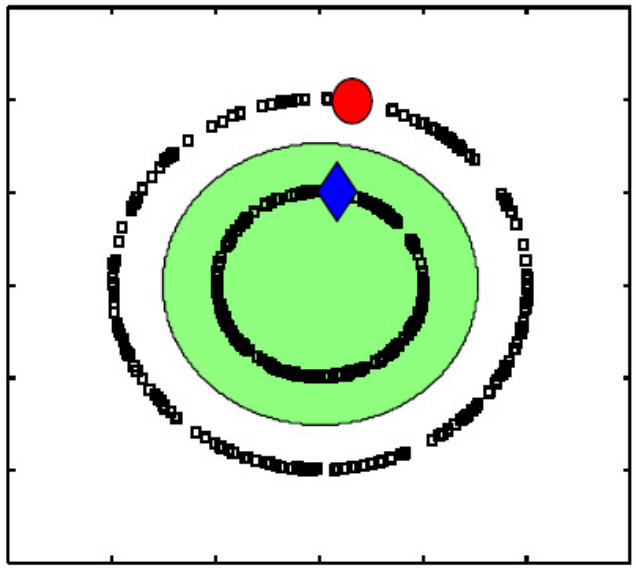
Laplacian SVM

LapSVM

- LapSVM integrates three concepts in machine learning:
  - Spectral graph theory
  - Manifold learning
  - Regularization in reproducing kernel Hilbert spaces (RKHS)

- Prior belief about the appropriate choice of classification functions can be influenced by the presence of unlabeled data:





Geometric Assumption

- Labeled examples:
  $(x, y) \in \mathcal{X} \times \mathbb{R}$ drawn according to some distribution P.
- Unlabeled examples:
  $x \in \mathcal{X}$ drawn according to the marginal distribution $P_{\mathcal{X}}$ of P.

- Assumption: If two points $x_1, x_2 \in \mathcal{X}$ are close in the intrinsic geometry of $P_{\mathcal{X}}$, then the conditional distributions $P(y|x_1)$ and $P(y|x_2)$ are similar.

In other words, the conditional distribution $P(y|x)$ varies smoothly along the geodesics in the intrinsic geometry of $P_{\mathcal{X}}$.

Standard Regularization Framework

- Let $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ be a Mercer kernel and $\mathcal{H}_k$ be the associated RKHS of functions $\mathcal{X} \rightarrow \mathbb{R}$ with norm $\|\cdot\|_k$.
- Given a set of labeled examples $\{(x_i, y_i)\}_{i=1}^n$ and a loss function V.
- Regularization framework for finding an optimal f^*:

$$f^* = \arg\min_{f \in \mathcal{H}_k} \left\{ \frac{1}{l} \sum_{i=1}^{l} V(x_i, y_i, f) + \gamma \|f\|_k^2 \right\}.$$

- Representer Theorem:
  The optimal solution exists in $\mathcal{H}_k$ and can be expressed as

$$f^*(x) = \sum_{i=1}^{l} \alpha_i k(x_i, x),$$

  for some real coefficients α_i. *Implication:* optimization can be performed over a finite-dimensional space of coefficients.

Extending the Standard Regularization Framework

- The manifold regularization approach extends the standard regularization framework by incorporating additional information about the geometric structure of the marginal distribution $P_{\mathcal{X}}$ into the regularized functional.
- The goal is to ensure that the solution is smooth w.r.t. both the ambient space and the marginal distribution $P_{\mathcal{X}}$.
- Since the additional regularizer depends on data, it can be called a data-dependent regularizer.

Extended Regularized Functional

- Extended regularized functional:

$$f^* = \arg\min_{f \in \mathcal{H}_k} \left\{ \frac{1}{l} \sum_{i=1}^{l} V(x_i, y_i, f) + \gamma_A \|f\|_k^2 + \gamma_I \|f\|_I^2 \right\},$$

  where γ_A controls the complexity of the function in the ambient space and γ_I controls the complexity of the function in the intrinsic geometry of $P_{\mathcal{X}}$.

- The additional regularizer should represent some penalty term that reflects the intrinsic structure of $P_{\mathcal{X}}$.

Graph Laplacian

- The manifold regularization term can be approximated using the graph Laplacian associated with the data.
- We construct an undirected, symmetric adjacency graph with $n = l + m$ nodes corresponding to the l labeled and m unlabeled examples, with $\mathbf{W} = (W_{ij})$ being the edge weights.
- Let $\mathbf{f} = (f(x_1), \ldots, f(x_n))^T$, $\mathbf{D} = (D_{ij})_{n \times n}$ where

$$D_{ij} = \begin{cases} \sum_{j=1}^{n} W_{ij} & i = j \\ 0 & i \neq j \end{cases},$$

and

$$\mathbf{L} = \mathbf{D} - \mathbf{W}$$

is called the graph Laplacian.

Graph Laplacian

- Consider this penalty measure:

$$\frac{1}{2}\sum_{i,j=1}^{n}(f(x_i)-f(x_j))^2 W_{ij}$$

$$\begin{aligned}
&= \frac{1}{2}\sum_{i,j=1}^{n} f(x_i)^2 W_{ij} + \frac{1}{2}\sum_{i,j=1}^{n} f(x_j)^2 W_{ij} - \sum_{i,j=1}^{n} f(x_i)f(x_j)W_{ij} \\
&= \sum_{i=1}^{n} f(x_i)^2 D_{ii} - \sum_{i,j=1}^{n} f(x_i)f(x_j)W_{ij} \\
&= \sum_{i,j=1}^{n} f(x_i)f(x_j)D_{ij} - \sum_{i,j=1}^{n} f(x_i)f(x_j)W_{ij} \\
&= \sum_{i,j=1}^{n} f(x_i)f(x_j)(D_{ij}-W_{ij}) = \mathbf{f}^T(\mathbf{D}-\mathbf{W})\mathbf{f} = \mathbf{f}^T\mathbf{L}\mathbf{f}.
\end{aligned}$$

- We use this measure to approximate $\|f\|_I^2$.

Empirical Estimation of New Regularization Term

- By incorporating the graph Laplacian, the optimization problem can be expressed as:

$$f^* = \arg\min_{f \in \mathcal{H}_k} \left\{ \frac{1}{l} \sum_{i=1}^{l} V(x_i, y_i, f) + \gamma_A \|f\|_k^2 + \frac{\gamma_I}{(l+m)^2} \mathbf{f}^T \mathbf{L} \mathbf{f} \right\}.$$

Representer Theorem for Optimization Problem based on Graph Laplacian

- Representer Theorem for optimization functional approximated using graph Laplacian:
  The minimizer of the optimization problem above admits an expansion of the following form:

$$f^*(x) = \sum_{i=1}^{l+m} \alpha_i k(x_i, x)$$

  in terms of both the labeled and unlabeled examples.

SVM

- Optimization problem with hinge loss function:

$$\min_{f \in \mathcal{H}_k} \left\{ \frac{1}{l} \sum_{i=1}^{l} (1 - y_i f(x_i))_+ + \gamma \|f\|_k^2 \right\},$$

where $(1 - yf(x))_+ = \max(0, 1 - yf(x))$ and $y_i \in \{-1, +1\}$.

- Optimal solution (from classical Representer Theorem):

$$f^*(x) = \sum_{i=1}^{l} \alpha_i^* k(x, x_i).$$

An unregularized bias terms b is often added to this form.

LapSVM

- Optimization problem:

$$\min_{f \in \mathcal{H}_k} \left\{ \frac{1}{l} \sum_{i=1}^{l} (1 - y_i f(x_i))_+ + \gamma_A \|f\|_k^2 + \frac{\gamma_I}{(l+m)^2} \mathbf{f}^T \mathbf{L} \mathbf{f} \right\}.$$

- Optimal solution (from new Representer Theorem):

$$f^*(x) = \sum_{i=1}^{l+m} \alpha_i^* k(x, x_i).$$

An unregularized bias terms b is often added to this form.

LapSVM

Advantage:

- LapSVM is among the very few graph-based SSL methods that can support inductive learning or out-of-sample extension (as opposed to transductive learning) in a principled way.

Disadvantage:

- Like SVM, extension of LapSVM from two-class to multi-class classification is not straightforward.

Discriminant Analysis

Discriminant Analysis

- Supervised dimensionality reduction:
  Using label information to obtain a low-dimensional representation of the data to facilitate the subsequent classification task (possibly using a very simple classifier such as nearest neighbor (1-NN) classifier).

- Linear methods:
  Linear discriminant analysis (LDA) and variants

- Nonlinear methods:
  Quadratic discriminant analysis (QDA), kernel discriminant analysis (KDA), etc.

LDA/KDA versus SVM

- LDA and KDA (kernel extension of LDA) have demonstrated successes in many classification applications, with performance (esp. for KDA) often comparable with that of SVM.
- LDA/KDA works for multi-class classification in the same way as for two-class classification.
- Optimization problem of LDA/KDA is more straightforward than that of SVM.
- Small sample size (SSS) problem for LDA/KDA:
  Within-class scatter matrix becomes singular when sample size is smaller than feature dimensionality, e.g., face recognition, text classification, microarray gene expression classification.

Linear Discriminant Analysis

LDA Basics

- Training set $\mathcal{D} = \{x_1, \ldots, x_n\}$, with $x_i \in \mathbb{R}^N$.
- $\mathcal{D}$ partitioned into $C \geq 2$ disjoint classes Π_i, with n_i examples in Π_i.
- Between-class and within-class scatter matrices:

$$S_b = \sum_{k=1}^{C} n_k (\bar{m}_k - \bar{m})(\bar{m}_k - \bar{m})^T$$

$$S_w = \sum_{k=1}^{C} \sum_{x_i \in \Pi_k} (x_i - \bar{m}_k)(x_i - \bar{m}_k)^T.$$

- LDA finds optimal projection matrix W^*:

$$W^* = \arg\max_W \mathrm{trace}((W^T S_w W)^{-1} W^T S_b W),$$

which can be computed from the eigenvectors of $S_w^{-1} S_b$.

Optimal Solution for LDA

- We use this alternative optimality criterion to find the (equivalent) optimal solution:

$$W^* = \arg\max_W \mathrm{trace}((W^T S_t W)^{-1} W^T S_b W),$$

  where $S_t = S_b + S_w$ is the total scatter matrix.

- A relevant theorem [Fukunaga, 1991]:

Theorem

For $W \in \mathbb{R}^{N \times (C-1)}$,

$$\max_W \mathrm{trace}((W^T S_t W)^{-1} W^T S_b W) = \mathrm{trace}(S_t^{-1} S_b).$$

Motivations for Our Work

Motivations

- Our work may be seen as killing two birds with one stone.

- Semi-supervised discriminant analysis (SSDA):
  We alleviate the SSS problem of LDA by exploiting unlabeled data, hence providing it with a semi-supervised extension.

Previous Work

Previous Work on Semi-Supervised Discriminant Analysis

- Like LDA, formulated as a generalized eigenvalue problem.
- Using unlabeled data to define an additional regularizer.
- SDA [Cai et al., ICCV 2007],
  SSLDA [Song et al., PR 2008],
  Semi-supervised LFDA [Sugiyama et al., PAKDD 2008],
  SSDA [Zhang and Yeung, CVPR 2008]

Semi-Supervised Discriminant Analysis via CCCP

Joint work with PhD student Yu Zhang
ECML PKDD 2008

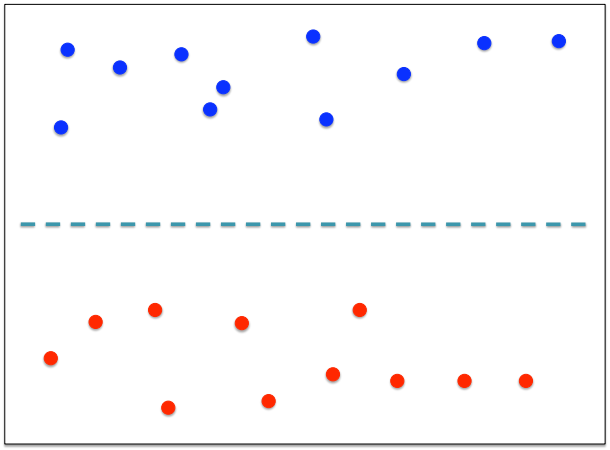
Notations for SSL Problem

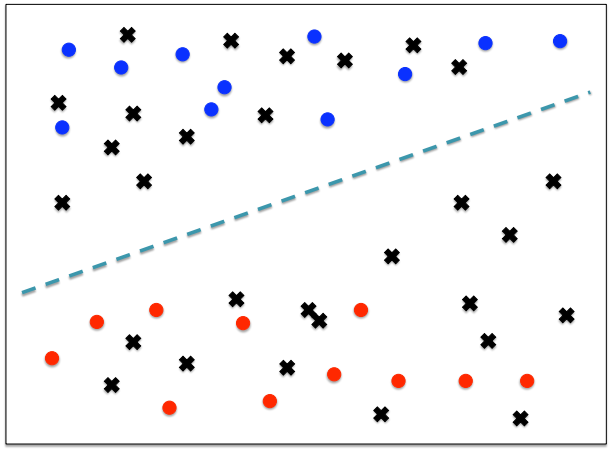
- l labeled data points $x_1, \ldots, x_l \in \mathbb{R}^N$ from C classes.
- m unlabeled data points $x_{l+1}, \ldots, x_{l+m} \in \mathbb{R}^N$ with unknown class labels (usually $l \ll m$).
- Training set has $n = l + m$ examples in total.

SSDA$_{CCCP}$

Optimality Criterion for SSDA$_{CCCP}$

- Inspired by TSVM [Joachims, 1999], we use unlabeled data to maximize the optimality criterion of LDA.





- From the theorem, the optimal criterion value of LDA is $\mathrm{trace}(S_t^{-1}S_b)$.
- So we utilize unlabeled data to maximize $\mathrm{trace}(S_t^{-1}S_b)$ via estimating the class labels of the unlabeled data points.

Optimality Criterion for SSDA$_{CCCP}$

- Class indicator matrix $A \in \mathbb{R}^{n \times C}$ with elements:

$$A_{ij} = \begin{cases} 1 & \text{if } x_i \in \Pi_j \\ 0 & \text{otherwise} \end{cases}$$

- Calculation of $\mathrm{trace}(S_t^{-1} S_b)$ from A:

$$\mathrm{trace}(S_t^{-1} S_b) = \sum_{k=1}^{C} \frac{1}{n_k} \left(A_k^T - \frac{n_k}{n} 1_n^T \right) S \left(A_k - \frac{n_k}{n} 1_n \right),$$

  where A_k is kth column of A, $S = D^T S_t^{-1} D$, and D is data matrix.

- Since the entries in A for the unlabeled data points are unknown, we maximize $\mathrm{trace}(S_t^{-1} S_b)$ w.r.t. A.

Constrained Optimization Problem

$$\max_{A,B_k,t_k} \quad \sum_{k=1}^{C} \frac{B_k^T S B_k}{t_k}$$

$$s.t. \quad t_k = A_k^T 1_n, \ k = 1, \ldots, C$$

$$B_k = A_k - \frac{t_k}{n} 1_n, \ k = 1, \ldots, C$$

$$A_{ij} = \begin{cases} 1 & \text{if } x_i \in \Pi_j \\ 0 & \text{otherwise} \end{cases} \quad i = 1, \ldots, l$$

$$A_{ij} \in \{0, 1\}, \ i = l+1, \ldots, n, \ j = 1, \ldots, C$$

$$\sum_{j=1}^{C} A_{ij} = 1, \ i = l+1, \ldots, n.$$

Constrained Optimization Problem

$$\max_{A,B_k,t_k} \quad \sum_{k=1}^{C} \frac{B_k^T S B_k}{t_k}$$

$$s.t. \quad t_k = A_k^T 1_n, \ k = 1, \ldots, C$$

$$B_k = A_k - \frac{t_k}{n} 1_n, \ k = 1, \ldots, C$$

$$A_{ij} = \begin{cases} 1 & \text{if } x_i \in \Pi_j \\ 0 & \text{otherwise} \end{cases} \quad i = 1, \ldots, l$$

$$A_{ij} \in \{0, 1\}, \ i = l+1, \ldots, n, \ j = 1, \ldots, C$$

$$\sum_{j=1}^{C} A_{ij} = 1, \ i = l+1, \ldots, n.$$

Constrained Optimization Problem

$$\max_{A,B_k,t_k} \quad \sum_{k=1}^{C} \frac{B_k^T S B_k}{t_k}$$

$$s.t. \quad t_k = A_k^T 1_n, \ k = 1, \ldots, C$$

$$B_k = A_k - \frac{t_k}{n} 1_n, \ k = 1, \ldots, C$$

$$A_{ij} = \begin{cases} 1 & \text{if } x_i \in \Pi_j \\ 0 & \text{otherwise} \end{cases} \quad i = 1, \ldots, l$$

$$A_{ij} \in \{0, 1\}, \ i = l{+}1, \ldots, n, \ j = 1, \ldots, C$$

$$\sum_{j=1}^{C} A_{ij} = 1, \ i = l{+}1, \ldots, n.$$

Constrained Optimization Problem

$$\max_{A,B_k,t_k} \quad \sum_{k=1}^{C} \frac{B_k^T S B_k}{t_k}$$

$$s.t. \quad t_k = A_k^T 1_n, \ k = 1, \ldots, C$$

$$B_k = A_k - \frac{t_k}{n} 1_n, \ k = 1, \ldots, C$$

$$A_{ij} = \begin{cases} 1 & \text{if } x_i \in \Pi_j \\ 0 & \text{otherwise} \end{cases} \quad i = 1, \ldots, l$$

$$A_{ij} \in \{0, 1\}, \ i = l{+}1, \ldots, n, \ j = 1, \ldots, C$$

$$\sum_{j=1}^{C} A_{ij} = 1, \ i = l{+}1, \ldots, n.$$

Constrained Optimization Problem

$$\max_{A,B_k,t_k} \quad \sum_{k=1}^{C} \frac{B_k^T S B_k}{t_k}$$

$$s.t. \quad t_k = A_k^T 1_n, \ k = 1, \ldots, C$$

$$B_k = A_k - \frac{t_k}{n} 1_n, \ k = 1, \ldots, C$$

$$A_{ij} = \begin{cases} 1 & \text{if } x_i \in \Pi_j \\ 0 & \text{otherwise} \end{cases} \quad i = 1, \ldots, l$$

$$A_{ij} \geq 0, \ i = l{+}1, \ldots, n, \ j = 1, \ldots, C$$

$$\sum_{j=1}^{C} A_{ij} = 1, \ i = l{+}1, \ldots, n.$$

Constrained Concave-Convex Procedure (CCCP)

- CCCP, closely related to difference of convex (DC) methods in optimization, is used to solve this non-convex optimization problem.

- Cost function $J(\theta)$ expressed as sum of convex and concave parts:

$$J(\theta) = J_{\text{vex}}(\theta) + J_{\text{cav}}(\theta)$$

- Each iteration of CCCP approximates $J_{\text{cav}}(\theta)$ by its tangent and minimizes the resulting convex function.

- Algorithm:

Initialize $\theta^{(0)}$ with a best guess.
repeat
$\theta^{(p)} = \arg\min_\theta \left(J_{\text{vex}}(\theta) + J'_{\text{cav}}(\theta^{(p-1)}) \cdot \theta \right)$
until convergence of $\theta^{(p)}$.

Constrained Concave-Convex Procedure (CCCP)

- From
$$\theta^{(p)} = \arg\min_{\theta} \left(J_{\text{vex}}(\theta) + J'_{\text{cav}}(\theta^{(p-1)}) \cdot \theta \right)$$
we get
$$J_{\text{vex}}(\theta^{(p)}) + J'_{\text{cav}}(\theta^{(p-1)}) \cdot \theta^{(p)} \leq J_{\text{vex}}(\theta^{(p-1)}) + J'_{\text{cav}}(\theta^{(p-1)}) \cdot \theta^{(p-1)}. \quad (1)$$
- From the concavity of $J_{\text{cav}}(\theta)$ we get
$$J_{\text{cav}}(\theta^{(p)}) \leq J_{\text{cav}}(\theta^{(p-1)}) + J'_{\text{cav}}(\theta^{(p-1)}) \cdot (\theta^{(p)} - \theta^{(p-1)}). \quad (2)$$
- Summing (1) and (2), we can show that $J(\theta^{(p)})$ decreases monotonically after each iteration:
$$J(\theta^{(p)}) \leq J(\theta^{(p-1)}).$$
- Still valid when θ is subject to constraints.

Constrained Concave-Convex Procedure (CCCP)

- Optimization problem in the $(p+1)$th iteration:

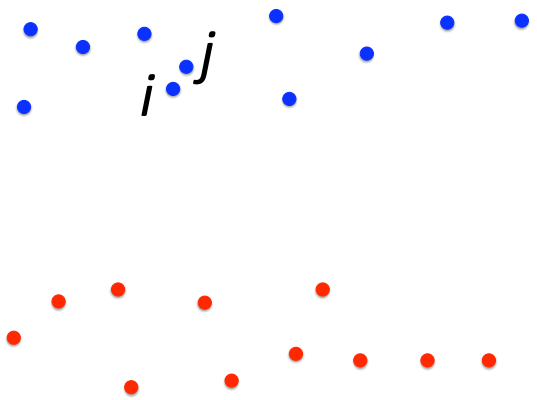
$$\begin{aligned}
\max_{A,B_k,t_k} \quad & \sum_{k=1}^{C}\left(\frac{2(B_k^{(p)})^T S}{t_k^{(p)}}B_k - \frac{(B_k^{(p)})^T S B_k^{(p)}}{(t_k^{(p)})^2}t_k\right) \\
s.t. \quad & t_k = A_k^T 1_n,\ k=1,\ldots,C \\
& B_k = A_k - \frac{t_k}{n}1_n,\ k=1,\ldots,C \\
& A_{ij} = \begin{cases} 1 & \text{if } x_i \in \Pi_j \\ 0 & \text{otherwise} \end{cases} \quad i=1,\ldots,l \\
& A_{ij} \geq 0,\ i=l+1,\ldots,n,\ j=1,\ldots,C \\
& \sum_{j=1}^{C} A_{ij} = 1,\ i=l+1,\ldots,n,
\end{aligned}$$

where $B_k^{(p)}, t_k^{(p)}$ were obtained in the pth iteration.

M-SSDA$_{CCCP}$

Manifold Assumption for M-SSDA$_{CCCP}$

- Manifold assumption:



$$\Rightarrow \quad A(i) \approx A(j)$$

- Given $\mathcal{D} = \{x_1, \ldots, x_n\}$, we construct a K-nearest neighbor graph $G = (V, E)$.
- Each edge is assigned a weight w_{ij}:

$$w_{ij} = \begin{cases} \exp\left(-\frac{\|x_i - x_j\|^2}{\sigma_i \sigma_j}\right) & \text{if } x_i \in N_K(x_j) \text{ or } x_j \in N_K(x_i) \\ 0 & \text{otherwise} \end{cases}$$

Optimization Problem for M-SSDA$_{CCCP}$

- Optimization problem for M-SSDA$_{CCCP}$:

$$\max_{A,B_k,t_k} \quad \sum_{k=1}^{C} \frac{B_k^T S B_k}{t_k} - \boxed{\lambda \sum_{i=1}^{n} \sum_{j=i+1}^{n} w_{ij} \|A(i) - A(j)\|_1}$$

$$s.t. \quad t_k = A_k^T 1_n, \ k = 1, \ldots, C$$

$$B_k = A_k - \frac{t_k}{n} 1_n, \ k = 1, \ldots, C$$

$$A_{ij} = \begin{cases} 1 & \text{if } x_i \in \Pi_j \\ 0 & \text{otherwise} \end{cases} \quad i = 1, \ldots, l$$

$$A_{ij} \geq 0, \ i = l{+}1, \ldots, n, \ j = 1, \ldots, C$$

$$\sum_{j=1}^{C} A_{ij} = 1, \ i = l{+}1, \ldots, n,$$

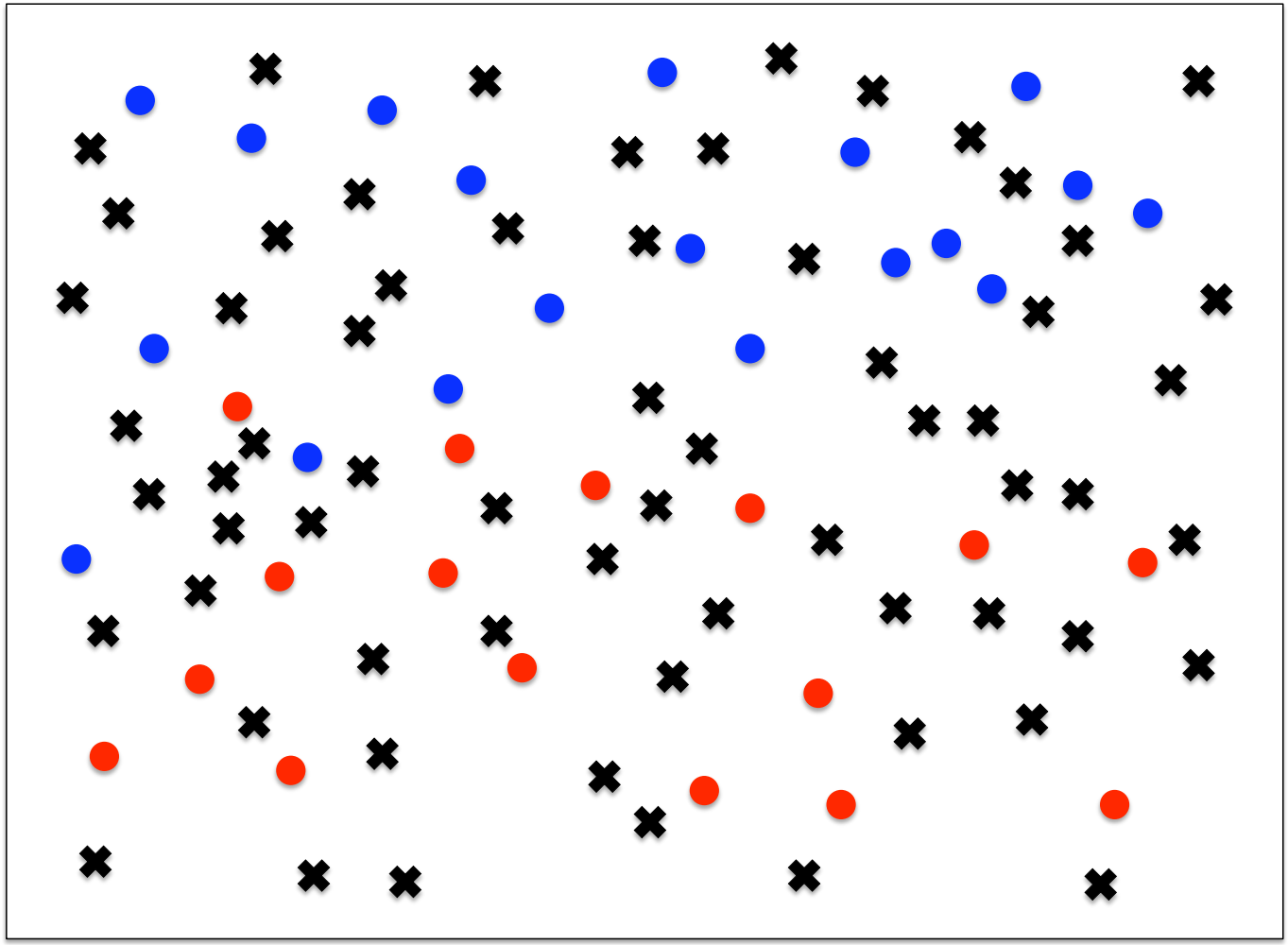
- This optimization problem can also be solved by CCCP.

Augmenting Labeled Data Set with Unlabeled Data

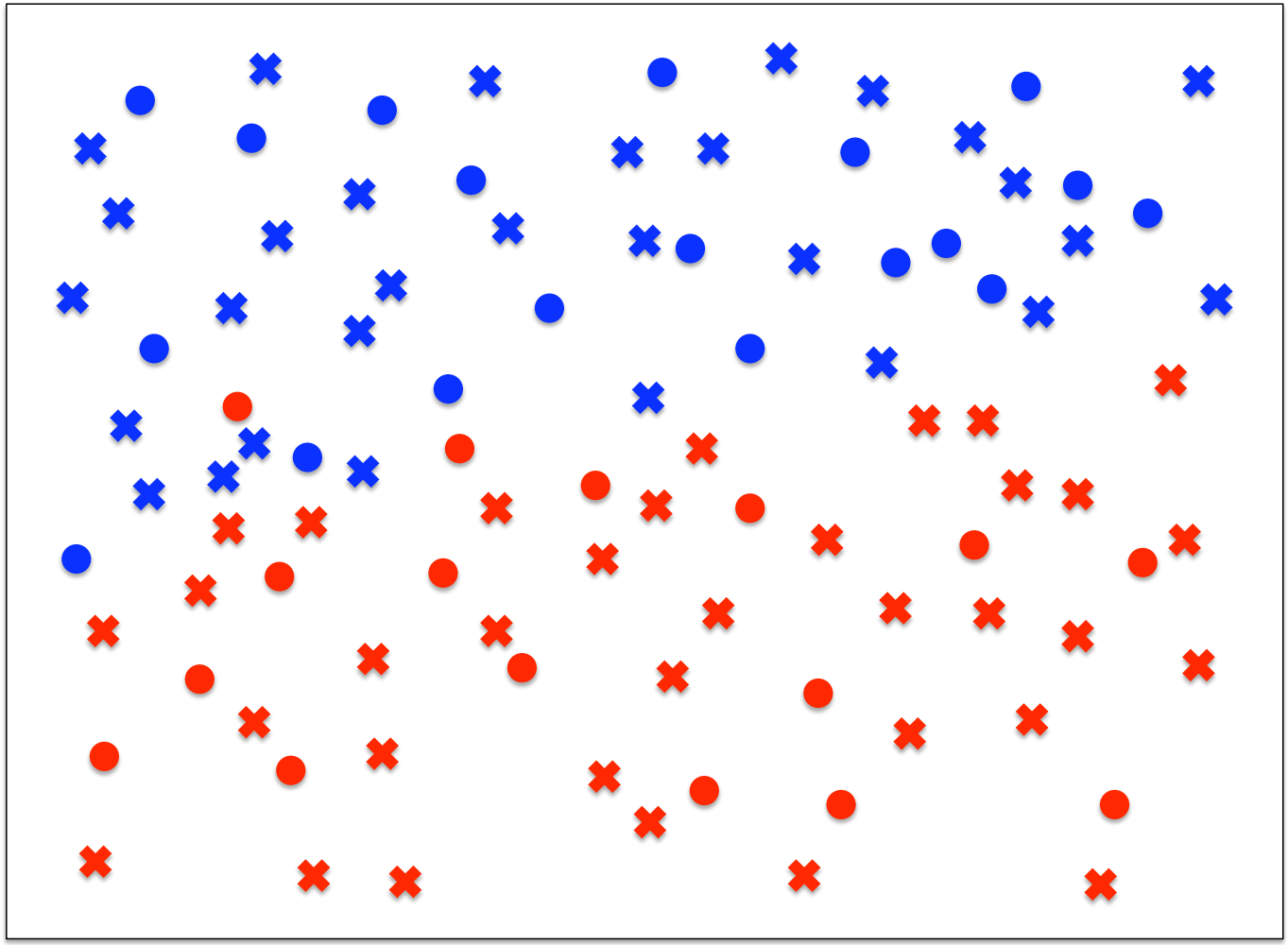
Augmenting Labeled Data Set with Unlabeled Data

- While solving the optimization problem, estimation of class labels for the unlabeled data is simultaneously performed.
- Not all the class labels can be estimated accurately.
- We propose a selection scheme for selecting unlabeled data points with reliably estimated class labels.

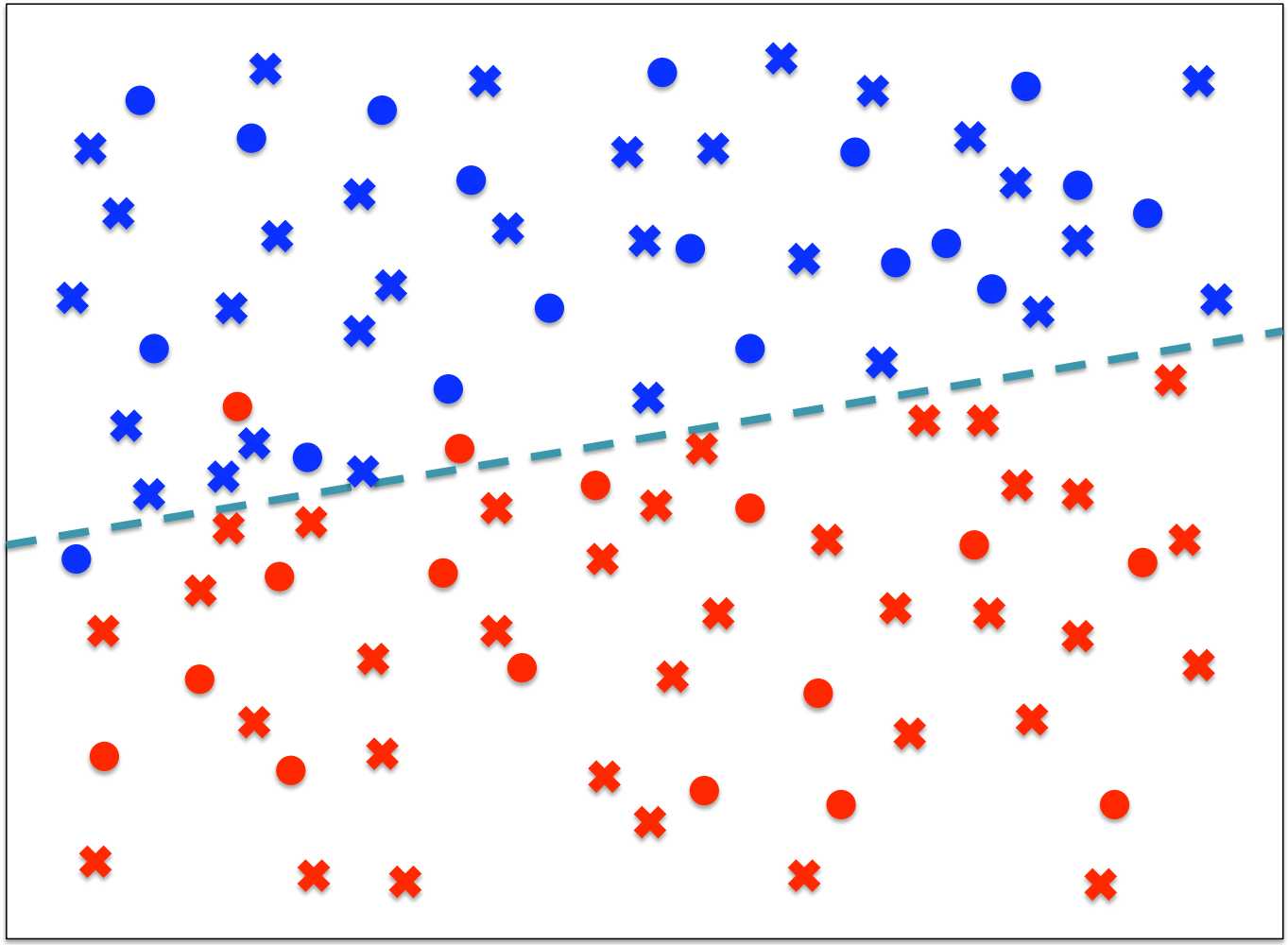
Data Selection Scheme



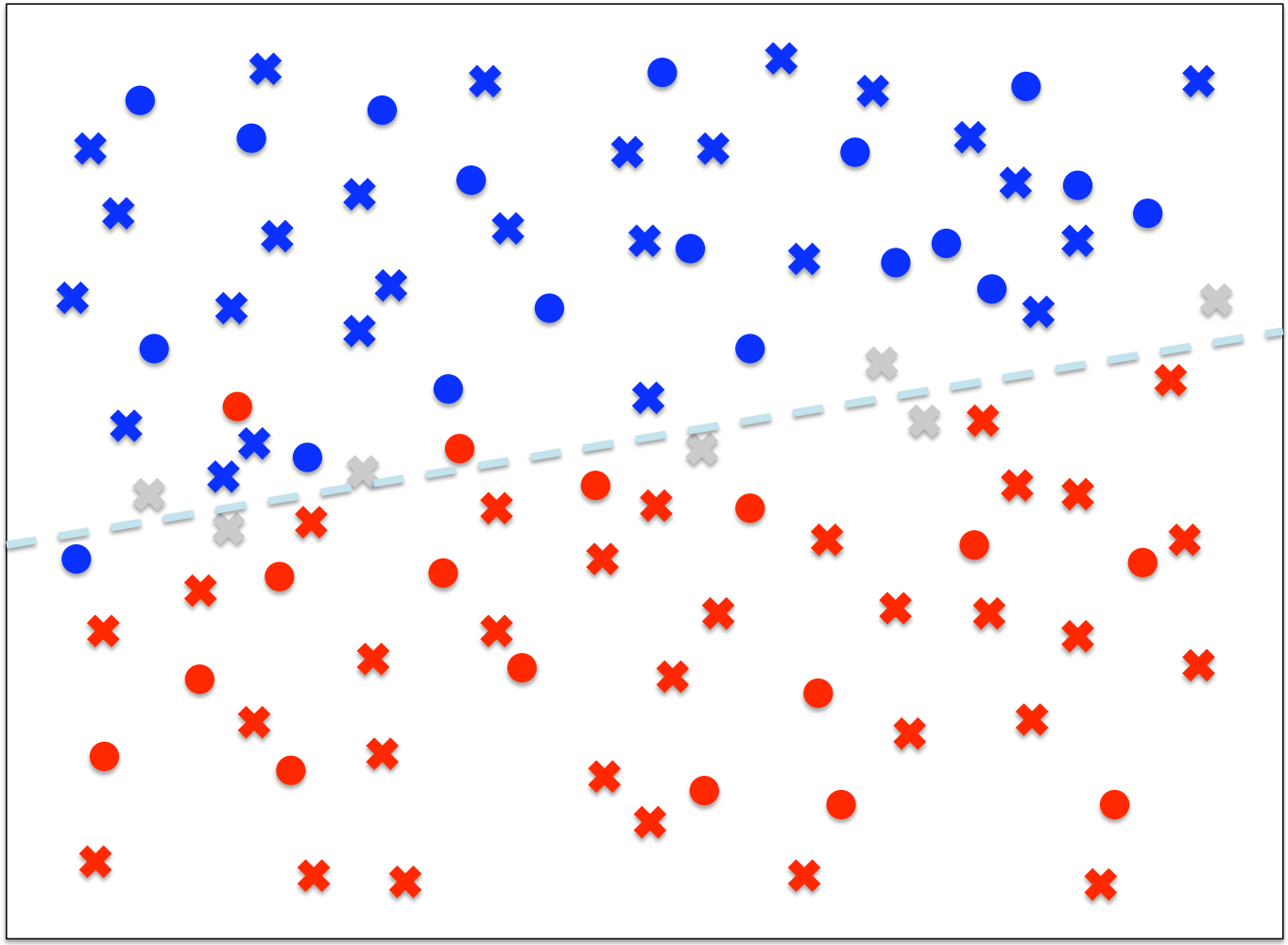
Data Selection Scheme



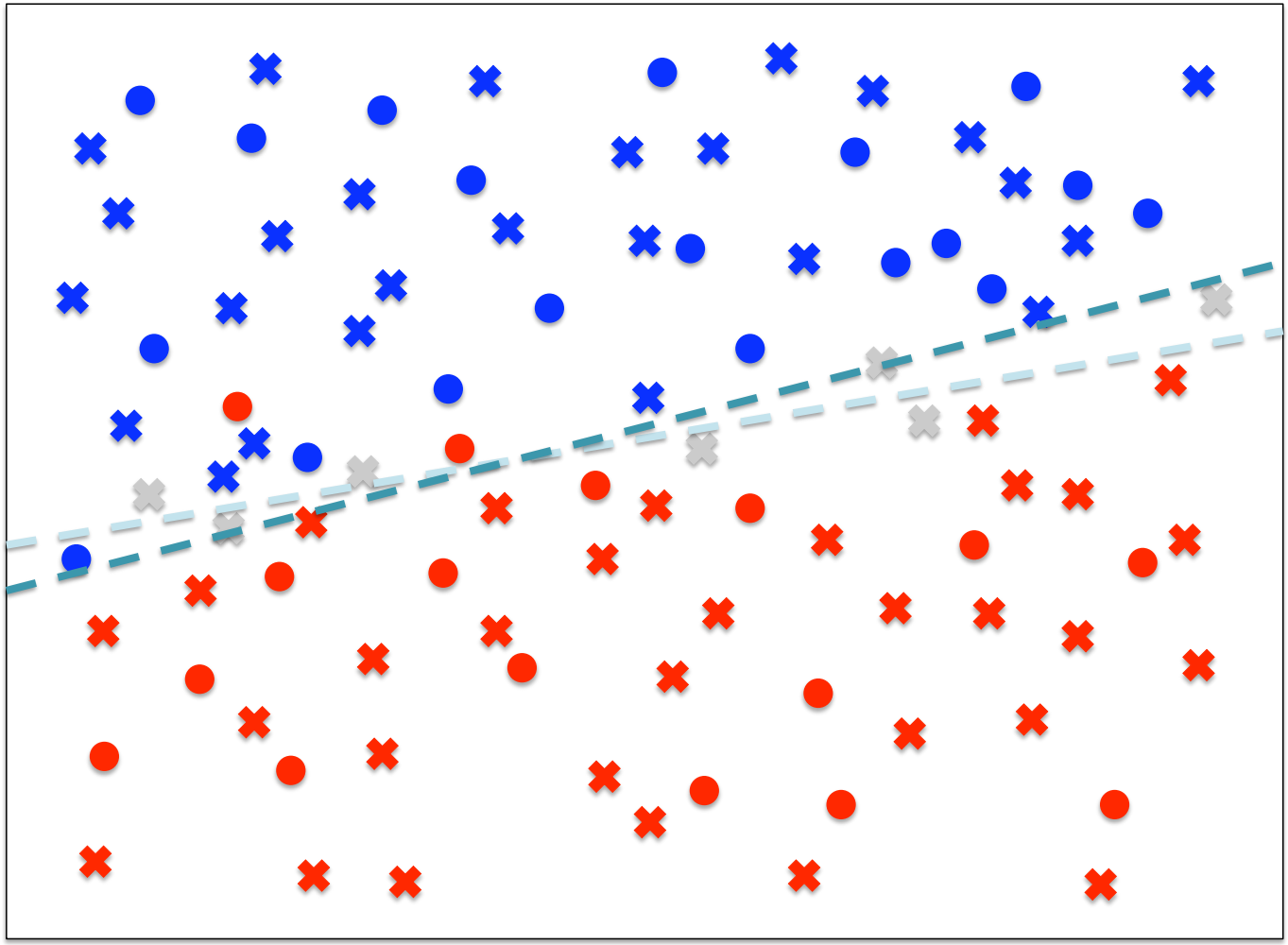
Data Selection Scheme



Data Selection Scheme



Data Selection Scheme



Computational Considerations

- Computation cost of SSDA$_{CCCP}$ and M-SSDA$_{CCCP}$ includes:
  - Performing LDA twice – $O(N^3)$ complexity
  - Solving the optimization problem using CCCP
- The linear programming (LP) problem inside each iteration of CCCP can be solved efficiently.
- In our experiments, CCCP converges very fast in less than 10 iterations.

Algorithm

Input: labeled data x_i $(i = 1, \ldots, l)$, unlabeled data x_i $(i = l+1, \ldots, n)$, K, θ, ε

Initialize $A^{(0)}$;

Initialize $B_k^{(0)}$ and $t_k^{(0)}$ based on $A^{(0)}$ for $k = 1, \ldots, C$;

Construct the K-nearest neighbor graph;

$p = 0$;

Repeat

$p = p + 1$;

Solve the optimization problem of SSDA$_{CCCP}$ or M-SSDA$_{CCCP}$;

Update $A^{(p)}$, $B_k^{(p)}$ and $t_k^{(p)}$ using the result of the optimization problem;

Until $\|A^{(p)} - A^{(p-1)}\|_F \leq \varepsilon$

Select the unlabeled data points with high confidence based on the threshold θ;

Augment the labeled data set and perform LDA to get W.

Output: transformation W

Experiments

Experimental Setup

- 11 benchmark data sets:
  - 8 UCI data sets
  - A brain-computer interface dataset (BCI)
  - Two image data sets (COIL and PIE)

- For each data set, we randomly select q data points from each class as labeled data and r points from each class as unlabeled data. The remaining data form the test set.
- For each partitioning, we perform 20 random splits and report the mean and standard derivation over the 20 trials.

Comparison with Other Dimensionality Reduction Methods

- Dimensionality reduction methods compared: PCA, PCA+LDA, SDA.
- Overall performance:

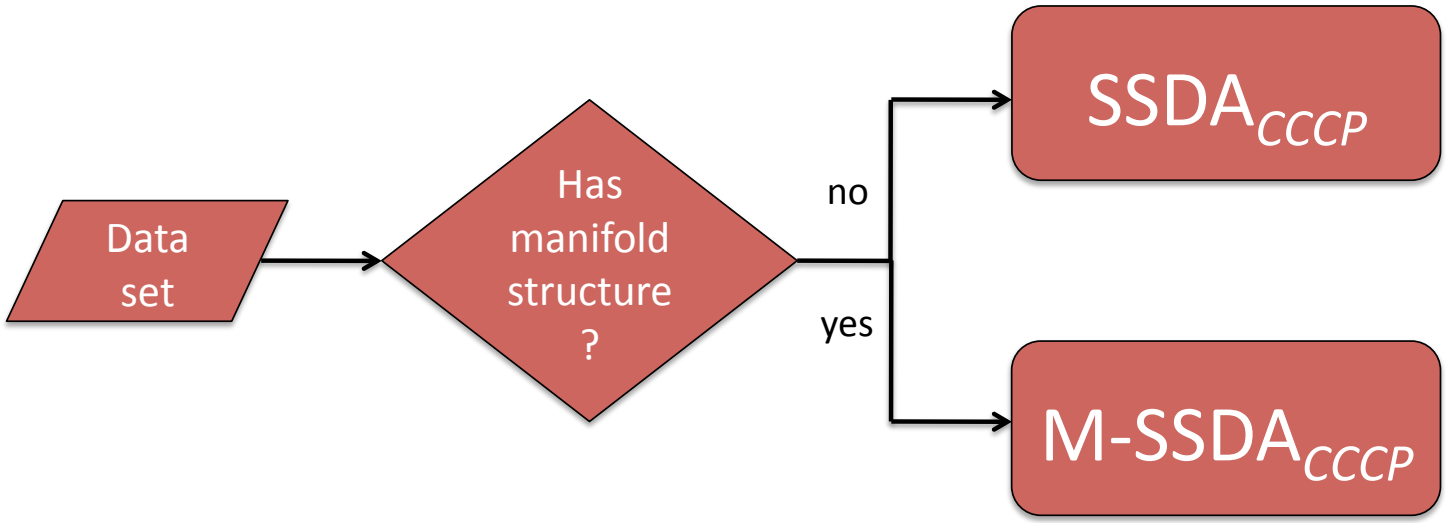
$$\{ \text{SSDA}_{CCCP}, \text{M-SSDA}_{CCCP} \} \geq \{ \text{PCA, PCA+LDA, SDA} \}$$

- Improvement is significant for DIABETES, HEART-STATLOG, PENDIGITS, VEHICLE and PIE.

Comparison with Other Dimensionality Reduction Methods

Data set	PCA	LDA	SDA	SSDA$_{CCCP}$	M-SSDA$_{CCCP}$
diabetes	0.4335(0.0775)	0.4438(0.0878)	0.4022(0.0638)	**0.3898(0.0674)**	0.4360(0.0605)
	0.4253(0.1154)	0.4311(0.0997)	0.3763(0.0864)	**0.3276(0.0643)**	0.4125(0.1074)
heart-statlog	0.4288(0.0689)	0.3978(0.0582)	0.3680(0.0564)	**0.3293(0.0976)**	0.3818(0.0662)
	0.3975(0.0669)	0.3767(0.1055)	0.3783(0.1076)	**0.3133(0.1174)**	0.3258(0.1493)
ionosphere	0.2895(0.1032)	0.2850(0.0876)	**0.2695(0.1056)**	0.2860(0.1015)	0.2830(0.1029)
	0.2189(0.0632)	0.2365(0.0972)	0.2241(0.0863)	0.2351(0.1032)	0.2399(0.1278)
hayes-roth	0.5175(0.0571)	0.4942(0.0531)	0.5058(0.0661)	0.4867(0.0569)	**0.4758(0.0586)**
	0.5115(0.0605)	0.5165(0.0690)	0.5077(0.0752)	0.5121(0.0770)	**0.5060(0.0627)**
iris	0.0917(0.0417)	0.0933(0.0613)	0.0825(0.0506)	0.0708(0.0445)	**0.0667(0.0493)**
	0.0907(0.0333)	0.0833(0.0586)	0.0809(0.0395)	0.0611(0.0370)	**0.0611(0.0454)**
mfeat-pixel	0.1450(0.0232)	0.1501(0.0290)	0.2783(0.0435)	0.1501(0.0289)	**0.1367(0.0210)**
	0.1429(0.0228)	0.1486(0.0264)	0.3428(0.0298)	0.1485(0.0264)	**0.1329(0.0213)**
pendigits	0.1724(0.0305)	0.2238(0.0364)	0.2547(0.0447)	0.1785(0.0266)	**0.1617(0.0242)**
	0.1761(0.0276)	0.2192(0.0332)	0.2544(0.0382)	0.1779(0.0190)	**0.1650(0.0225)**
vehicle	0.5739(0.0375)	0.5741(0.0365)	0.5400(0.0402)	**0.4396(0.0734)**	0.4838(0.0901)
	0.5808(0.0453)	0.5879(0.0429)	0.5462(0.0312)	**0.4329(0.0672)**	0.4739(0.0791)
BCI	0.4835(0.0460)	0.4830(0.0557)	0.4960(0.0476)	**0.4750(0.0432)**	0.4975(0.0484)
	0.5000(0.0324)	0.4803(0.0249)	0.4812(0.0326)	**0.4732(0.0331)**	0.4741(0.0346)
COIL	**0.4443(0.0418)**	0.5247(0.0371)	0.5419(0.0607)	0.5236(0.0374)	0.5193(0.0401)
	0.4391(0.0364)	0.5194(0.0421)	0.5461(0.04821)	0.5178(0.0434)	0.5096(0.0398)
PIE	0.6156(0.0275)	0.5055(0.1624)	0.7629(0.0377)	0.4674(0.1757)	**0.2381(0.0552)**
	0.6207(0.0251)	0.5126(0.1512)	0.8277(0.0208)	0.4777(0.1696)	**0.2424(0.0592)**

SSDA$_{CCCP}$ or M-SSDA$_{CCCP}$?



Effectiveness of Selection Method

- Mean accuracy of label estimation for unlabeled data over 20 trials before and after applying the selection method:

Data set	SSDA_{CCCP} (%)		M-SSDA_{CCCP} (%)	
	Before	After	Before	After
diabetes	64.03	66.67	54.10	51.20
heart-statlog	72.27	72.62	55.25	66.70
ionosphere	69.05	87.51	74.10	82.07
hayes-roth	46.75	52.73	42.00	42.64
iris	75.42	93.39	91.42	95.06
mfeat-pixel	32.49	100.0	94.21	98.91
pendigits	75.31	86.08	88.92	94.02
vehicle	56.30	69.88	44.80	52.26
BCI	50.75	65.42	49.00	49.15
COIL	33.57	96.07	42.64	60.03
PIE	30.48	85.00	52.64	70.41

Comparison with Graph-Based SSL Methods

- Graph-based SSL methods compared: LapSVM, LapRLS.
- Same experimental settings as before.
- Overall performance:

$$\{ \text{SSDA}_{CCCP}, \text{M-SSDA}_{CCCP} \} \geq \{ \text{LapSVM, LapRLS} \}$$

- One advantage of SSDA_{CCCP} and M-SSDA_{CCCP}:
  Same formulation and optimization procedure for two-class and multi-class problems.

Comparison with Graph-Based SSL Methods

Data set	LapSVM	LapRLS	SSDA$_{CCCP}$	M-SSDA$_{CCCP}$
diabetes	0.4763(0.0586)	0.4523(0.0650)	**0.3620(0.0680)**	0.4015(0.0893)
	0.5643(0.0684)	0.5009(0.0775)	**0.3488(0.0514)**	0.4234(0.1107)
heart-statlog	0.3478(0.1059)	0.3348(0.1070)	**0.3108(0.0901)**	0.3758(0.0914)
	0.3517(0.1458)	0.3375(0.1366)	**0.3091(0.0989)**	0.3442(0.1226)
ionosphere	0.3525(0.0539)	0.3260(0.0527)	0.3340(0.0902)	**0.3185(0.0719)**
	0.2245(0.0697)	0.2266(0.0732)	0.2705(0.0969)	0.2905(0.0933)
hayes-roth	0.6633(0.0149)	0.6608(0.0261)	**0.4833(0.0824)**	0.5225(0.0466)
	0.5550(0.0737)	0.5500(0.0516)	**0.4901(0.0705)**	0.5104(0.0711)
iris	0.3175(0.1390)	0.2708(0.1474)	0.0650(0.0516)	**0.0525(0.0437)**
	0.3049(0.1426)	0.2741(0.1473)	0.0772(0.0508)	**0.0593(0.0379)**
mfeat-pixel	0.1488(0.0236)	**0.1359(0.0257)**	0.1578(0.0268)	0.1420(0.0249)
	0.2252(0.0187)	0.2075(0.0181)	0.1555(0.0263)	**0.1427(0.0183)**
pendigits	0.2571(0.0379)	0.2368(0.0312)	0.1856(0.0226)	**0.1697(0.0245)**
	0.2539(0.0334)	0.2377(0.0283)	0.1866(0.0244)	**0.1735(0.0217)**
vehicle	0.4713(0.0449)	0.4921(0.0460)	**0.4219(0.0623)**	0.4645(0.0770)
	0.4758(0.0477)	0.5007(0.0452)	**0.4181(0.0600)**	0.4641(0.0777)
BCI	0.4805(0.0551)	0.4695(0.0612)	**0.4515(0.0543)**	0.4665(0.0479)
	0.4631(0.0456)	**0.4562(0.0390)**	0.4752(0.0362)	0.4864(0.0372)
COIL	0.5414(0.0496)	0.5855(0.0617)	**0.5028(0.0576)**	0.5030(0.0488)
	0.5421(0.0497)	0.5864(0.0598)	**0.5057(0.0533)**	0.5062(0.0423)
PIE	0.2561(0.0311)	0.3405(0.0227)	0.4096(0.1600)	**0.2497(0.0313)**
	0.2671(0.0235)	0.3523(0.0151)	0.4160(0.1575)	**0.2556(0.0235)**

Conclusion

Conclusion

- In this work, we have proposed a semi-supervised extension to LDA, which also allows it to alleviate the small sample size problem.

- Possible future work:
  - Kernel extensions to deal with nonlinearity
  - Semi-supervised extensions of other dimensionality reduction methods.