Learning From Data Locally and Globally

Kaizhu Huang

Supervisors:
Prof. Irwin King,
Prof. Michael R. Lyu

Outline

- Background
  - Linear Binary classifier
  - Global Learning
    - Bayes optimal Classifier
  - Local Learning
    - Support Vector Machine
- Contributions
- Minimum Error Minimax Probability Machine (MEMPM)
  - Biased Minimax Probability Machine (BMPM)
- Maxi-Min Margin Machine (M^4)
  - Local Support Vector Regression (LSVR)
- Future work
- Conclusion

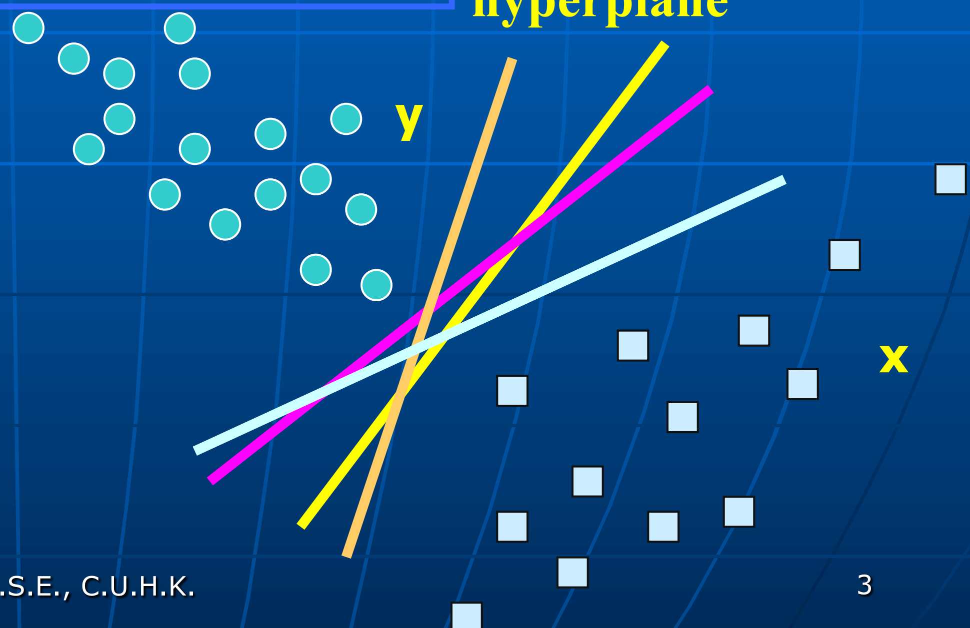
Background - Linear Binary Classifier

Given two classes of data sampled from **x** and **y**, we are trying to find a linear decision plane **wT z + b=0**, which can correctly discriminate **x** from **y**.

wT z + b< 0, **z** is classified as **y**;

wT z + b >0, **z** is classified as **x**.

w^T z + b=0 : decision hyperplane



Background - Global Learning (I)

- Global learning
  - **Basic idea**: Focusing on summarizing data usually by estimating a distribution
  - **Example**
    - 1) Assume Gaussinity for the data
    - 2) Learn the parameters via MLE or other criteria
    - 3) Exploit Bayes theory to find the optimal thresholding for classification

Traditional Bayes Optimal Classifier

Background - Global Learning (II)

- Problems
  - I Usually have to assume specific models on data, which may NOT always coincide with data

    "all models are wrong but some are useful…"—by George Box

  - II Estimating distributions may be wasteful and imprecise

    Finding the ideal generator of the data, i.e., the distribution, is only an intermediate goal in many settings, e.g., in classification or regression. ***Optimizing an intermediate objective may be inefficient or wasteful****.*

Background- Local Learning (I)

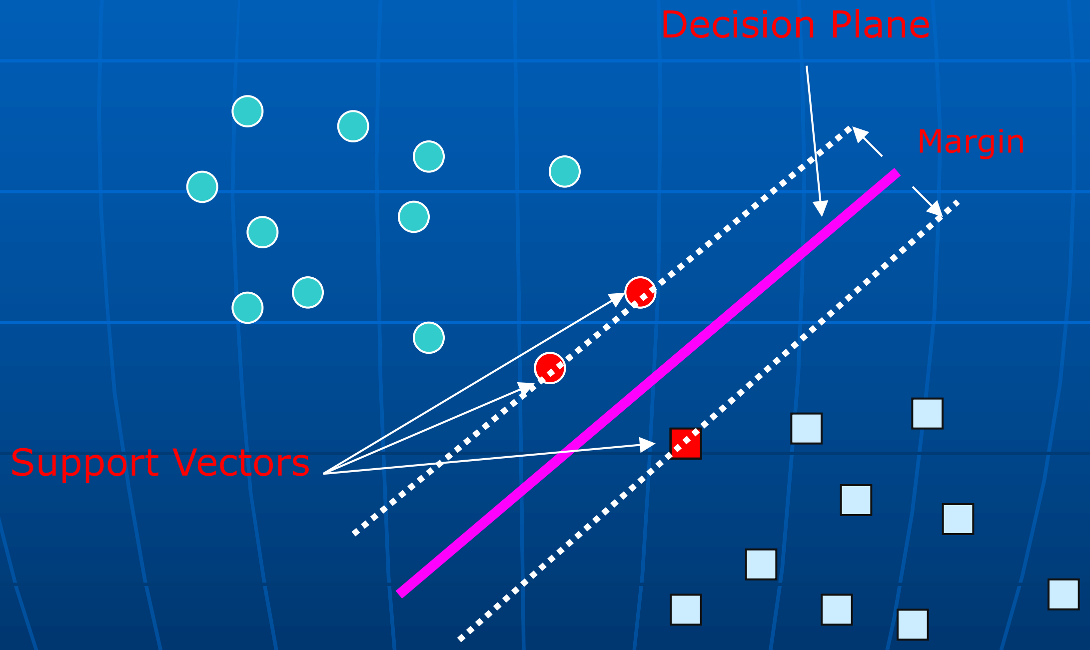
- Local learning
  - **Basic idea**: Focus on exploiting part of information, which is directly related to the objective, e.g., the classification accuracy instead of describing data in a holistic way
  - **Example**

    In classification, we need to accurately model the data around the (possible) separating plane, while inaccurate modeling of other parts is certainly acceptable (as is done in SVM).

Background - Local Learning (II)

- Support Vector Machine (SVM)

  ---The current state-of-the-art classifier



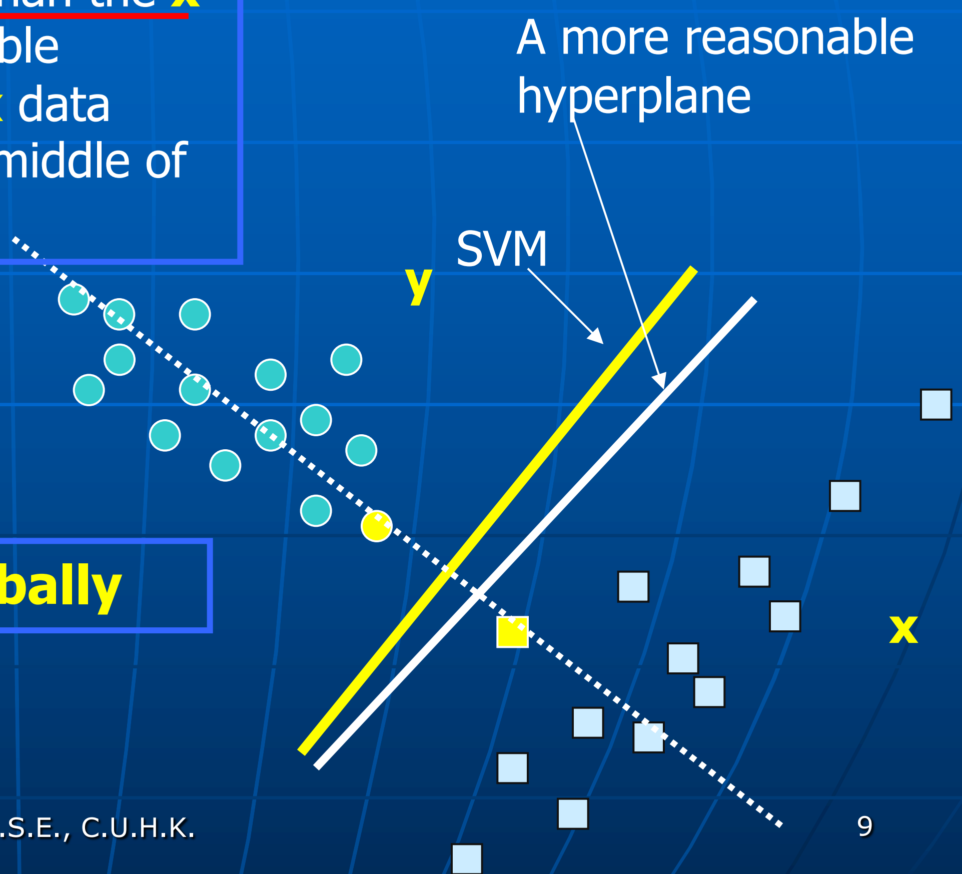
Background - Local Learning (III)

- Problems
  - The fact that the objective is exclusively determined by local information may lose the overall view of data

Background- Local Learning (IV)

An illustrative example

Along the dashed axis, the **y** data is obviously more likely to scatter than the **x** data. Therefore, a more reasonable hyerplane may lie closer to the **x** data rather than locating itself in the middle of two classes as in SVM.



Learning Locally and Globally

Learning Locally and Globally

- **Basic idea**: Focus on using both local information and certain robust global information
  - Do not try to estimate the distribution as in global learning, which may be inaccurate and indirect
  - Consider robust global information for providing a roadmap for local learning

Summary of Background

Global Learning

Problem II *Optimizing an intermediate objective*

Can we directly optimize the objective??

Local Learning, e.g., SVM

Problem I *Assume specific models*

Without specific model assumption?

Distribution-free Bayes optimal classifier ---
Minimum Error Minimax Probability Machine (MEMPM)

Local Learning

Problem III *Focusing on local info may lose the roadmap of data*

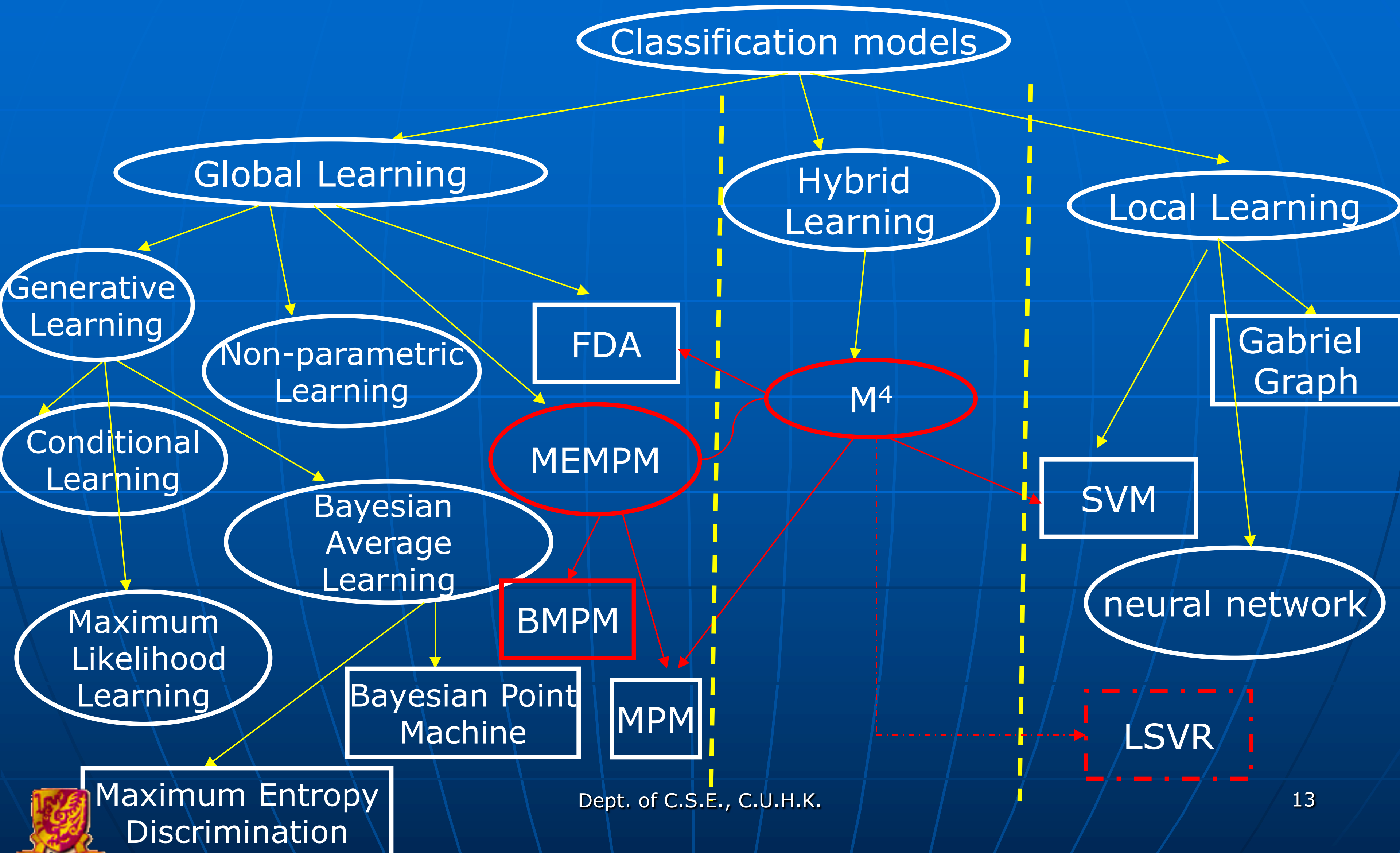
Can we learn both globally and locally??

Maxi-Min Margin Machine (M^4)

Contributions

- Mininum Error Minimax Probability Machine
  (Accepted by JMLR 04)
  - A worst-case distribution-free Bayes Optimal Classifier
  - Containing Minimax Probability Machine (MPM) and Biased Minimax Probability Machine (BMPM)*(AMAI04,CVPR04)* as special cases
- Maxi-Min Margin Machine (M^4) (*ICML 04+Submitted*)
  - A unified framework that learns locally and globally
    - Support Vector Machine (SVM)
    - Minimax Probability Machine (MPM)
    - Fisher Discriminant Analysis (FDA)
    - Can be linked with MEMPM
  - Can be extended into regression: Local Support Vector Regression (LSVR) *(submitted)*

Hierarchy Graph of Related Models



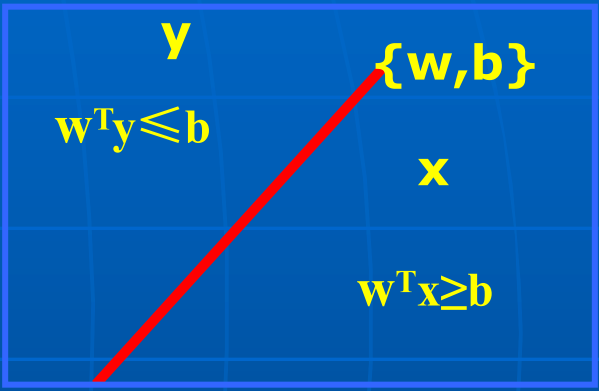
Minimum Error Minimax Probability Machine (MEMPM)

Model Definition:

$$\max_{\alpha,\beta,\mathbf{w}\neq\mathbf{0},b} \quad \theta\alpha+(1-\theta)\beta, \quad \text{s.t.}$$

$$\inf_{\mathbf{x}\sim(\overline{\mathbf{x}},\Sigma_{\mathbf{x}})} \quad \Pr\{\mathbf{w}^T\mathbf{x}\geq b\}\geq\alpha,$$

$$\inf_{\mathbf{y}\sim(\overline{\mathbf{y}},\Sigma_{\mathbf{y}})} \quad \Pr\{\mathbf{w}^T\mathbf{y}\leq b\}\geq\beta.$$



- ***θ:*** prior probability of class **x**; *α(β):* represents the worst-case accuracy for class x (y)

$\mathbf{x}\sim(\overline{\mathbf{x}},\Sigma_{\mathbf{x}})$: The class of distributions that have prescribed mean $\overline{\mathbf{x}}$ and covariance $\Sigma_{\mathbf{x}}$

$\mathbf{y}\sim(\overline{\mathbf{y}},\Sigma_{\mathbf{y}})$: likewise

MEMPM: Model Comparison

MEMPM (*JMLR04*)

$$\max_{\alpha,\beta,\text{[redacted]}\neq\mathbf{0},b} \quad \theta\alpha+(1-\theta)\beta, \quad \text{s.t.}$$

$$\inf_{\mathbf{x}\sim(\bar{\mathbf{x}},\Sigma_{\mathbf{x}})} \quad \Pr\{\text{[redacted]}^T\mathbf{x}\geq b\}\geq\alpha,$$

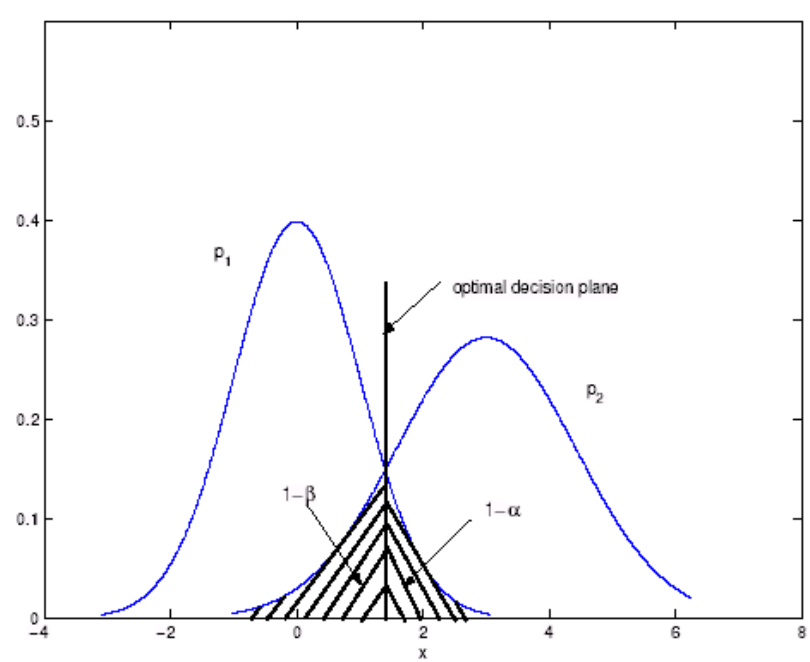
$$\inf_{\mathbf{y}\sim(\bar{\mathbf{y}},\Sigma_{\mathbf{y}})} \quad \Pr\{\text{[redacted]}^T\mathbf{y}\leq b\}\geq\beta.$$

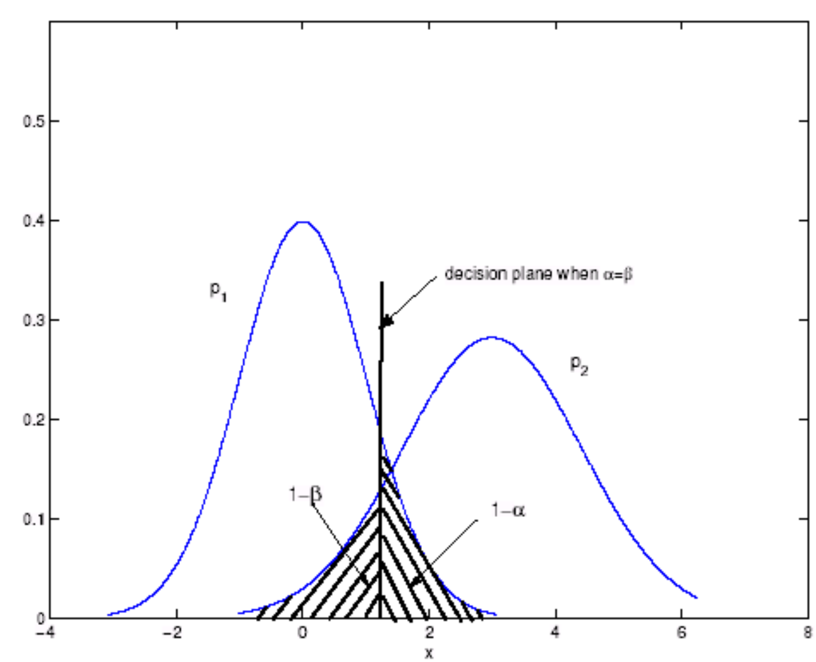
MPM (*Lanckriet et al. JMLR 2002*)

$$\max_{\alpha,\mathbf{w}\neq\mathbf{0},b} \alpha \quad \text{s.t.}$$

$$\inf_{\mathbf{y}\sim(\bar{\mathbf{x}},\Sigma_{\mathbf{x}})} \quad \Pr\{\mathbf{w}^T\mathbf{x}\geq b\}\geq\alpha$$

$$\inf_{\mathbf{y}\sim(\bar{\mathbf{y}},\Sigma_{\mathbf{y}})} \quad \Pr\{\mathbf{w}^T\mathbf{y}\leq b\}\geq\alpha$$





MEMPM: Advantages

- A **distribution-free** Bayes optimal Classifier in the worst-case scenario
- Containing an **explicit accuracy bound**, namely, $\theta\alpha + (1-\theta)\beta$
- Subsuming a special case *Biased Minimax Probability Machine* **for biased classification**

MEMPM: Biased MPM

Biased Classification:

Diagnosis of epidemical disease: Classifying a patient who is infected with a disease into an opposite class results in more serious consequence than the other way around.

The classification accuracy should **be biased towards the class with disease.**

BMPM

$$
\begin{aligned}
&\max_{\alpha,\beta,\mathbf{w}\neq\mathbf{0},b} \quad \alpha, \quad \text{s.t.} \\
&\inf_{\mathbf{x}\sim(\bar{\mathbf{x}},\Sigma_{\mathbf{x}})} \Pr\{\mathbf{w}^T\mathbf{x}\geq b\}\geq\alpha, \\
&\inf_{\mathbf{y}\sim(\bar{\mathbf{y}},\Sigma_{\mathbf{y}})} \Pr\{\mathbf{w}^T\mathbf{y}\leq b\}\geq\beta, \\
&\beta\geq\gamma
\end{aligned}
$$

An ideal model for biased classification.

A typical setting: We should **maximize the accuracy** for the important class as long as the accuracy for **the less important class is acceptable** (greater than an acceptable level γ).

MEMPM: Biased MPM (I)

- Objective

$$\max_{\alpha,\beta,\mathbf{a}\neq\mathbf{0},b} \quad \alpha \quad \text{s.t.} \quad \inf_{\mathbf{x}\sim(\overline{\mathbf{x}},\Sigma_{\mathbf{x}})} \Pr\{\mathbf{a}^T\mathbf{x} \geq b\} \geq \alpha$$
$$\inf_{\mathbf{y}\sim(\overline{\mathbf{y}},\Sigma_{\mathbf{y}})} \Pr\{\mathbf{a}^T\mathbf{y} \leq b\} \geq \beta$$
$$\beta \geq \gamma$$

- Equivalently

$$\max_{\alpha,\beta,\mathbf{a}\neq\mathbf{0},b} \quad \kappa(\alpha) \quad \text{s.t.} \quad 1 = \kappa(\alpha)\sqrt{\mathbf{a}^T\Sigma_{\mathbf{x}}\mathbf{a}} + \kappa(\beta)\sqrt{\mathbf{a}^T\Sigma_{\mathbf{y}}\mathbf{a}}$$
$$\mathbf{a}^T(\overline{\mathbf{x}}-\overline{\mathbf{y}}) = 1$$
$$\kappa(\beta) \geq \kappa(\gamma)$$
$$\kappa(\alpha) = \sqrt{\tfrac{\alpha}{1-\alpha}}, \quad \kappa(\beta) = \sqrt{\tfrac{\beta}{1-\beta}}$$

MEMPM: Biased MPM (II)

- Objective

$$\max_{\alpha,\beta,\mathbf{w}\neq\mathbf{0},b} \kappa(\alpha) \quad \text{s.t.} \quad 1=\kappa(\alpha)\sqrt{\mathbf{w}^T\Sigma_{\mathbf{x}}\mathbf{w}}+\kappa(\beta)\sqrt{\mathbf{w}^T\Sigma_{\mathbf{y}}\mathbf{w}}$$
$$\mathbf{w}^T(\overline{\mathbf{x}}-\overline{\mathbf{y}})=1$$
$$\kappa(\beta)\geq\kappa(\gamma)$$

- Equivalently,

$$\max_{\kappa(\beta),\mathbf{w}\neq\mathbf{0}} \frac{1-\kappa(\beta)\sqrt{\mathbf{w}^T\Sigma_{\mathbf{y}}\mathbf{w}}}{\sqrt{\mathbf{w}^T\Sigma_{\mathbf{x}}\mathbf{w}}} \quad \text{s.t.} \quad \mathbf{w}^T(\overline{\mathbf{x}}-\overline{\mathbf{y}})=1, \quad \kappa(\beta)\geq\kappa(\gamma)$$

- Equivalently,

$$\max_{\mathbf{w}\neq\mathbf{0}} \frac{1-\kappa(\gamma)\sqrt{\mathbf{w}^T\Sigma_{\mathbf{y}}\mathbf{w}}}{\sqrt{\mathbf{w}^T\Sigma_{\mathbf{x}}\mathbf{w}}} \quad \text{s.t.} \quad \mathbf{w}^T(\overline{\mathbf{x}}-\overline{\mathbf{y}})=1$$

1. Each local optimum is the **global optimum**
2. Can be solved in ***$O(n^3+Nn^2)$***

Conave-Convex Fractional Programming problem

N: *number of data points* ***n:*** *Dimension*

MEMPM: Optimization (I)

- Objective

$$\max_{\alpha,\beta,\mathbf{w}\neq\mathbf{0}} \quad \theta\alpha+(1-\theta)\beta, \quad \text{s.t.}$$

$$1=\kappa(\alpha)\sqrt{\mathbf{w}^T\Sigma_{\mathbf{x}}\mathbf{w}}+\kappa(\beta)\sqrt{\mathbf{w}^T\Sigma_{\mathbf{y}}\mathbf{w}}$$

$$\mathbf{w}^T(\overline{\mathbf{x}}-\overline{\mathbf{y}})=1$$

- $$\min_{\kappa(\alpha),\kappa(\beta),\mathbf{w}\neq\mathbf{0}} \quad \frac{\theta}{\kappa(\alpha)^2+1}+\frac{1-\theta}{\kappa(\beta)^2+1}, \quad \text{s.t.}$$

$$1=\kappa(\alpha)\sqrt{\mathbf{w}^T\Sigma_{\mathbf{x}}\mathbf{w}}+\kappa(\beta)\sqrt{\mathbf{w}^T\Sigma_{\mathbf{y}}\mathbf{w}}$$

$$\mathbf{w}^T(\overline{\mathbf{x}}-\overline{\mathbf{y}})=1$$

MEMPM: Optimization (II)

- Objective

$$\max_{\alpha,\beta,\mathbf{w}\neq\mathbf{0}} \quad \frac{\theta\kappa(\alpha)^2}{\kappa(\alpha)^2+1}+(1-\theta)\beta, \quad \text{s.t.}$$

$$\mathbf{w}^T(\overline{\mathbf{x}}-\overline{\mathbf{y}})=1,$$

$$where \quad \kappa(\alpha)=\frac{1-\kappa(\beta)\sqrt{\mathbf{w}^T\Sigma_{\mathbf{y}}\mathbf{w}}}{\sqrt{\mathbf{w}^T\Sigma_{\mathbf{x}}\mathbf{w}}}$$

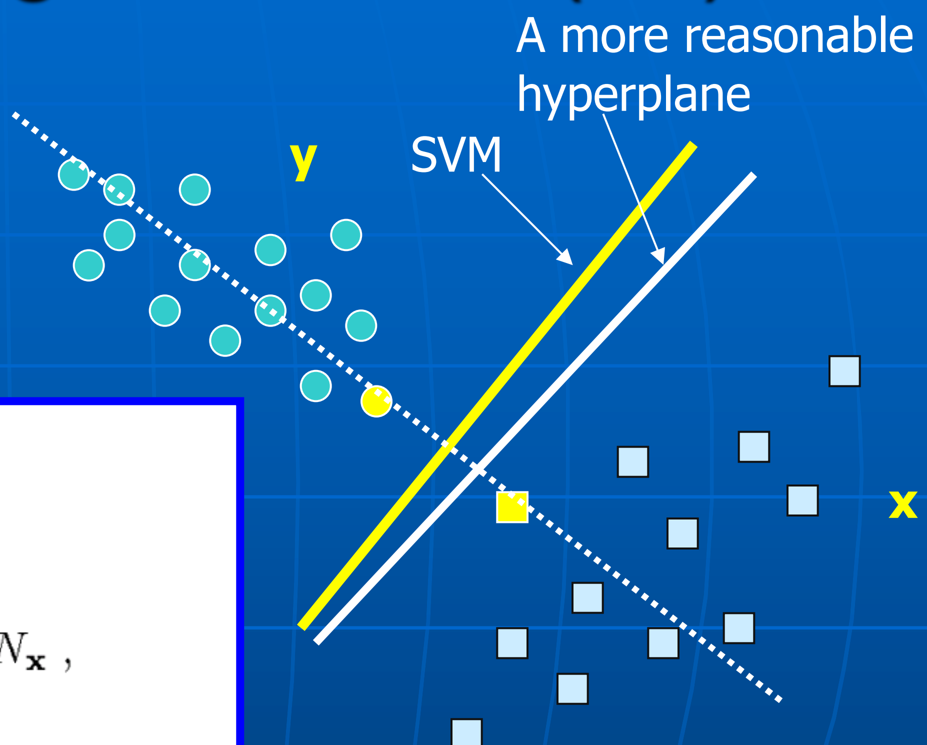
- Line search + BMPM method

MEMPM: Problems

- As a global learning approach, the decision plane is exclusively dependent on global information, i.e., up to second-order moments.
- These moments may NOT be accurately estimated! –We may need local information to neutralize the negative effect caused.

Learning Locally and Globally

Learning Locally and Globally: Maxi-Min Margin Machine (M^4)



Model Definition

The formulation for M^4 can be written as:

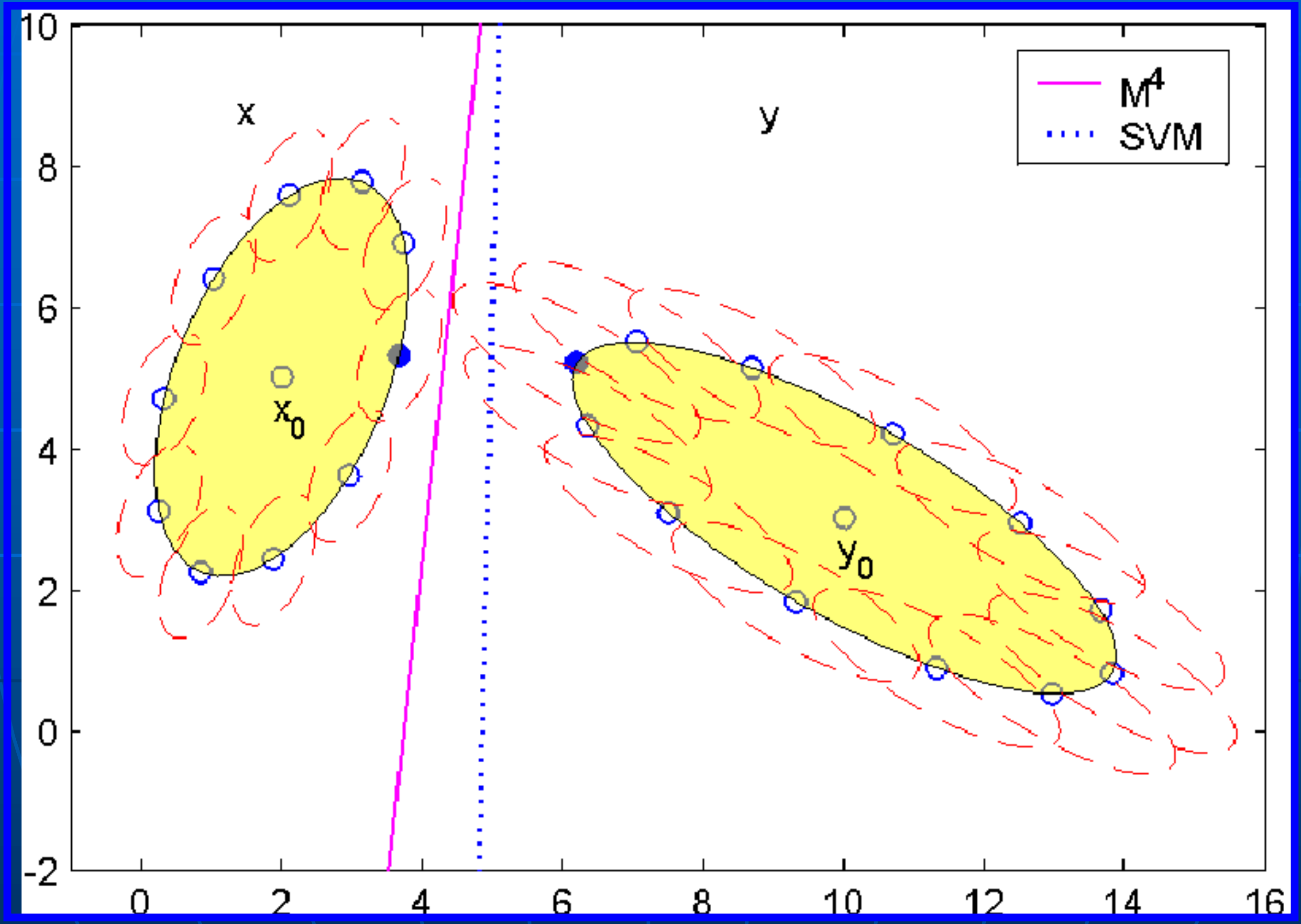
$$\max_{\rho, \mathbf{w} \neq \mathbf{0}, b} \quad \rho \quad s.t.$$

$$\frac{(\mathbf{w}^T \mathbf{x}_i + b)}{\sqrt{\mathbf{w}^T \Sigma_{\mathbf{x}} \mathbf{w}}} \geq \rho, \quad i = 1, 2, \ldots, N_{\mathbf{x}} ,$$

$$\frac{-(\mathbf{w}^T \mathbf{y}_j + b)}{\sqrt{\mathbf{w}^T \Sigma_{\mathbf{y}} \mathbf{w}}} \geq \rho, \quad j = 1, 2, \ldots, N_{\mathbf{y}} ,$$

where $\Sigma_{\mathbf{x}}$ and $\Sigma_{\mathbf{y}}$ refer to the covariance matrices of the $\mathbf{x}$ and the $\mathbf{y}$ data, respectively.

M^4: Geometric Interpretation



M^4: Solving Method (I)

Divide and Conquer:

If we fix ρ to a specific ρ_n , the problem changes to check whether this ρ_n satisfies the following constraints:

$$(\mathbf{w}^T\mathbf{x}_i + b) \geq \rho_n\sqrt{\mathbf{w}^T\Sigma_{\mathbf{x}}\mathbf{w}},\ i = 1, \ldots, N_{\mathbf{x}}\ ,$$
$$-(\mathbf{w}^T\mathbf{y}_j + b) \geq \rho_n\sqrt{\mathbf{w}^T\Sigma_{\mathbf{y}}\mathbf{w}},\ j = 1, \ldots, N_{\mathbf{y}}\ .$$

If yes, we increase ρ_n; otherwise, we decrease it.

Second Order Cone Programming Problem!!!

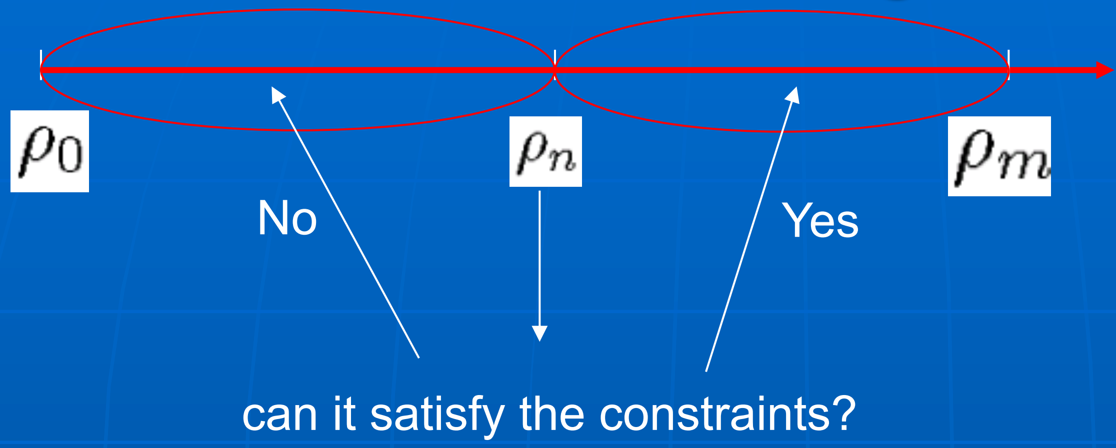
M^4: Solving Method (II)

Iterate the following two Divide and Conquer steps:

1. **Divide:** Set $\rho_n = (\rho_0 + \rho_m)/2$, where ρ_0 is a feasible ρ, ρ_m is an infeasible ρ, and $\rho_0 \leq \rho_m$.
2. **Conquer:** Call the Modified Second Order Cone Programming (MSOCP) procedure elaborated in the following to check whether ρ_n is a feasible ρ. If yes, set $\rho_0 = \rho_n$; otherwise, set $\rho_m = \rho_n$;

Sequential Second Order Cone Programming Problem!!!

M^4: Solving Method (III)



- The worst-case iteration number is ***log(L/ε)***

  L: $\rho_{max} - \rho_{min}$ (search range)

  ε : The required precision
- Each iteration is a Second Order Cone Programming problem yielding ***$O(n^3)$***
- Cost of forming the constraint matrix ***$O(N\, n^3)$***

  Total time complexity= ***$O(\log(L/\varepsilon)\, n^3 + N\, n^3) \approx O(N\, n^3)$***

  N: *number of data points* ***n:*** *Dimension*

M^4: Links with MPM (I)

$$\frac{(\mathbf{w}^T\mathbf{x}_i + b)}{\sqrt{\mathbf{w}^T\Sigma_{\mathbf{x}}\mathbf{w}}} \geq \rho, \quad i = 1, 2, \ldots, N_{\mathbf{x}} ,$$

$$\frac{-(\mathbf{w}^T\mathbf{y}_j + b)}{\sqrt{\mathbf{w}^T\Sigma_{\mathbf{y}}\mathbf{w}}} \geq \rho, \quad j = 1, 2, \ldots, N_{\mathbf{y}} ,$$

Span all the data points and add them together

$$\mathbf{w}^T\sum_{i=1}^{N_{\mathbf{x}}}\mathbf{x}_i + N_{\mathbf{x}}b \geq N_{\mathbf{x}}\rho\sqrt{\mathbf{w}^T\Sigma_{\mathbf{x}}\mathbf{w}} ,$$
$$\Leftrightarrow \quad \mathbf{w}^T\overline{\mathbf{x}} + b \geq \rho\sqrt{\mathbf{w}^T\Sigma_{\mathbf{x}}\mathbf{w}} ,$$

$$-(\mathbf{w}^T\sum_{j=1}^{N_{\mathbf{y}}}\mathbf{y}_j + N_{\mathbf{y}}b) \geq N_{\mathbf{y}}\rho\sqrt{\mathbf{w}^T\Sigma_{\mathbf{y}}\mathbf{w}} ,$$
$$\Leftrightarrow \quad -(\mathbf{w}^T\overline{\mathbf{y}} + b) \geq \rho\sqrt{\mathbf{w}^T\Sigma_{\mathbf{y}}\mathbf{w}} ,$$

+

$$\max_{\rho,\mathbf{w}\neq\mathbf{0},b} \quad \rho \quad s.t.$$
$$\mathbf{w}^T(\overline{\mathbf{x}} - \overline{\mathbf{y}}) \geq \rho(\sqrt{\mathbf{w}^T\Sigma_{\mathbf{x}}\mathbf{w}} + \sqrt{\mathbf{w}^T\Sigma_{\mathbf{y}}\mathbf{w}}) \quad (15)$$

Exactly MPM Optimization Problem!!!

M^4: Links with MPM (II)

Remarks:

- The procedure is not reversible: MPM is **a special case of M4**
- MPM focuses on building decision boundary **GLOBALLY**, i.e., it exclusively depends on the means and covariances.
- However, means and covariances may not be accurately estimated.

The formulation for M^4 can be written as:

$$\max_{\rho,\mathbf{w}\neq\mathbf{0},b} \quad \rho \quad s.t.$$

$$\frac{(\mathbf{w}^T\mathbf{x}_i + b)}{\sqrt{\mathbf{w}^T\Sigma_{\mathbf{x}}\mathbf{w}}} \geq \rho, \quad i = 1, 2, \ldots, N_{\mathbf{x}} \ ,$$

$$\frac{-(\mathbf{w}^T\mathbf{y}_j + b)}{\sqrt{\mathbf{w}^T\Sigma_{\mathbf{y}}\mathbf{w}}} \geq \rho, \quad j = 1, 2, \ldots, N_{\mathbf{y}} \ ,$$

where $\Sigma_{\mathbf{x}}$ and $\Sigma_{\mathbf{y}}$ refer to the covariance matrices of the $\mathbf{x}$ and the $\mathbf{y}$ data, respectively.

$$\max_{\rho,\mathbf{w}\neq\mathbf{0},b} \quad \rho \quad s.$$

$$\mathbf{w}^T(\overline{\mathbf{x}} - \overline{\mathbf{y}}) \geq \rho(\sqrt{\mathbf{w}^T\Sigma_{\mathbf{x}}\mathbf{w}} + \sqrt{\mathbf{w}^T\Sigma_{\mathbf{y}}\mathbf{w}})$$

M^4: Links with SVM (I)

1

If one assumes $\Sigma_{\mathbf{x}} = \Sigma_{\mathbf{y}} = \Sigma$,

2

$$\max_{\rho, \mathbf{w} \neq \mathbf{0}, b} \quad \rho \quad s.t.$$

$$(\mathbf{w}^T \mathbf{x}_i + b) \geq \rho \sqrt{\mathbf{w}^T \Sigma \mathbf{w}} \ ,$$

$$-(\mathbf{w}^T \mathbf{y}_j + b) \geq \rho \sqrt{\mathbf{w}^T \Sigma \mathbf{w}} \ ,$$

where $i = 1, \ldots, N_{\mathbf{x}}$ and $j = 1, \ldots$

The magnitude of w can scale up without influencing the optimization. Assume $\rho(\mathbf{w}^T \Sigma \mathbf{w})^{0.5}=1$

3

$$\min_{\mathbf{w} \neq \mathbf{0}, b} \quad \mathbf{w}^T \Sigma \mathbf{w} \quad s.t.$$

$$(\mathbf{w}^T \mathbf{x}_i + b) \geq 1 \ ,$$

$$-(\mathbf{w}^T \mathbf{y}_j + b) \geq 1 \ ,$$

where $i = 1, \ldots, N_{\mathbf{x}}$ and $j = 1, \ldots, N_{\mathbf{y}}$.

4

If one assumes Σ=I

Support Vector Machines

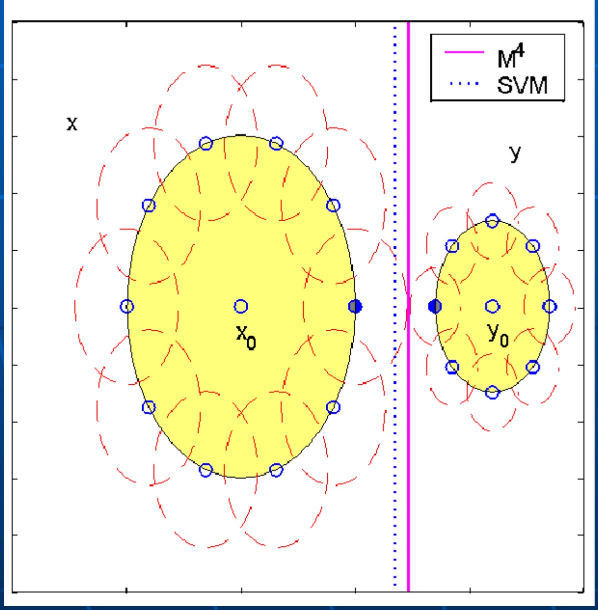
SVM is the special case of M^4

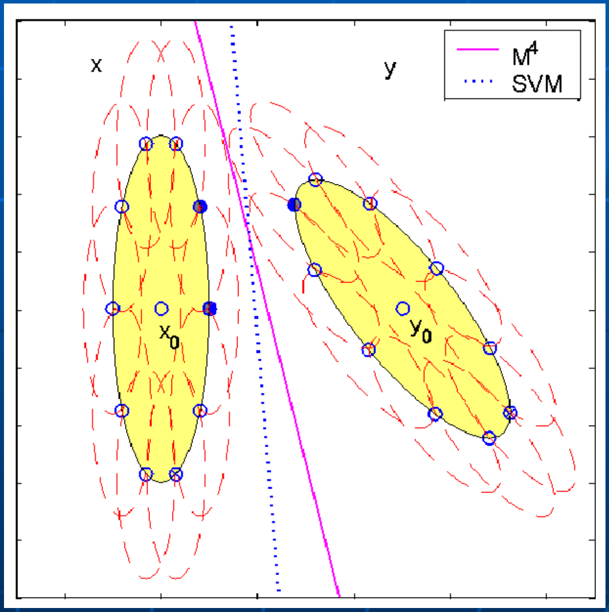
M^4: Links with SVM (II)

Assumption 1 → If one assumes $\Sigma_{\mathbf{x}} = \Sigma_{\mathbf{y}} = \Sigma$,

Assumption 2 → **If one assumes $\Sigma = I$**

These two assumptions of SVM are **inappropriate**





M^4: Links with FDA (I)

If one assumes
$\Sigma_x=\Sigma_y=(\Sigma^*_y+\Sigma^*_x)/2$

$$\max_{\rho,\mathbf{w}\neq\mathbf{0},b} \quad \rho \quad s.t.$$

$$\frac{(\mathbf{w}^T\mathbf{x}_i+b)}{\sqrt{\mathbf{w}^T\Sigma_{\mathbf{x}}\mathbf{w}+\mathbf{w}^T\Sigma_{\mathbf{y}}\mathbf{w}}} \geq \rho \ ,$$

$$\frac{-(\mathbf{w}^T\mathbf{y}_j+b)}{\sqrt{\mathbf{w}^T\Sigma_{\mathbf{x}}\mathbf{w}+\mathbf{w}^T\Sigma_{\mathbf{y}}\mathbf{w}}} \geq \rho \ ,$$

Perform a procedure similar to MPM…

FDA

$$\max_{\rho,\mathbf{w}\neq\mathbf{0},b} \quad \rho \quad s.t.$$

$$\mathbf{w}^T(\overline{\mathbf{x}}-\overline{\mathbf{y}}) \geq \rho\sqrt{\mathbf{w}^T\Sigma_{\mathbf{x}}\mathbf{w}+\mathbf{w}^T\Sigma_{\mathbf{y}}\mathbf{w}}$$

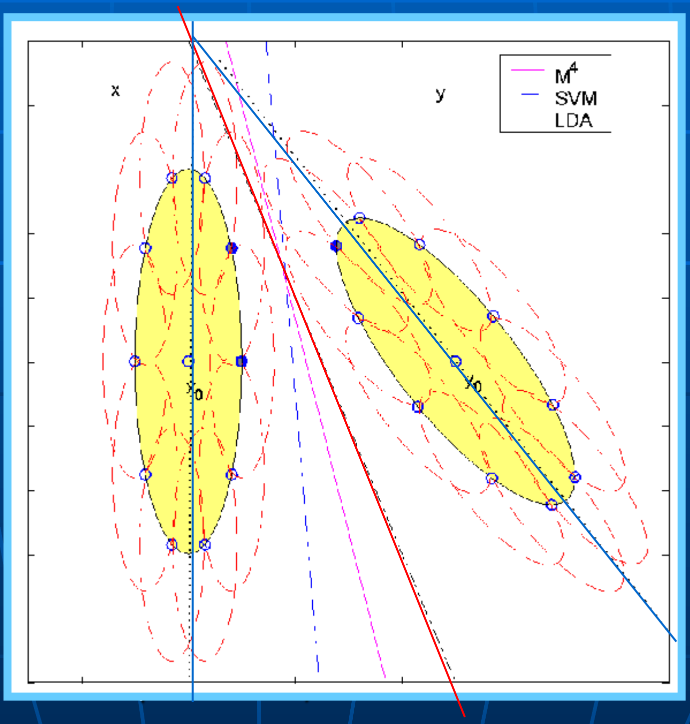
M^4: Links with FDA (II)

Assumption

If one assumes
$\Sigma x=\Sigma y=(\Sigma^{*}y+\Sigma^{*}x)/2$

?

Still inappropriate



M^4: Links with MEMPM

M^4 (a globalized version)

$$\max_{\mathbf{w}\neq 0, b} \quad \theta t + (1-\theta) s \quad \text{s.t.}$$
$$\frac{\mathbf{w}^T\overline{\mathbf{x}} + b}{\sqrt{\mathbf{w}^T\Sigma_{\mathbf{x}}\mathbf{w}}} \geq t\,,$$
$$-\frac{\mathbf{w}^T\overline{\mathbf{y}} + b}{\sqrt{\mathbf{w}^T\Sigma_{\mathbf{y}}\mathbf{w}}} \geq s\,,$$

MEMPM

$$\max_{\alpha,\beta,\mathbf{a}\neq 0, b} \quad \theta\alpha + (1-\theta)\beta \quad \text{s.t.}$$
$$\inf_{\mathbf{x}\sim(\overline{\mathbf{x}},\Sigma_{\mathbf{x}})} \Pr\{\mathbf{a}^T\mathbf{x} \geq b\} \geq \alpha,$$
$$\inf_{\mathbf{y}\sim(\overline{\mathbf{y}},\Sigma_{\mathbf{y}})} \Pr\{\mathbf{a}^T\mathbf{y} \leq b\} \geq \beta.$$

$$\max_{\alpha,\beta,\mathbf{a}\neq 0, b} \quad \theta\alpha + (1-\theta)\beta \quad \text{s.t.}$$
$$-b + \mathbf{a}^T\overline{\mathbf{x}} \geq \kappa(\alpha)\sqrt{\mathbf{a}^T\Sigma_{\mathbf{x}}\mathbf{a}}\,,$$
$$b - \mathbf{a}^T\overline{\mathbf{y}} \geq \kappa(\beta)\sqrt{\mathbf{a}^T\Sigma_{\mathbf{y}}\mathbf{a}}\,.$$

T and s ⟷ ***K(α) and K(β)*** :

The margin from the mean to the decision plane

The globalized M^4 **maximizes the weighted margin**, while MEMPM **Maximizes the weighted worst-case accuracy**.

M^4 :Nonseparable Case

Introducing slack variables

$$\max_{\rho, \mathbf{w} \neq \mathbf{0}, b, \boldsymbol{\xi}} \quad \rho - C \sum_{k=1}^{N_{\mathbf{x}}+N_{\mathbf{y}}} \boldsymbol{\xi}_k \quad s.t.$$

$$(\mathbf{w}^T \mathbf{x}_i + b) \geq \rho \sqrt{\mathbf{w}^T \Sigma_{\mathbf{x}} \mathbf{w}} - \boldsymbol{\xi}_i \ ,$$

$$-(\mathbf{w}^T \mathbf{y}_j + b) \geq \rho \sqrt{\mathbf{w}^T \Sigma_{\mathbf{y}} \mathbf{w}} - \boldsymbol{\xi}_{j+N_{\mathbf{x}}} \ ,$$

$$\xi_k \geq 0 \ ,$$

How to solve?? **Line Search+Second Order Cone Programming**

Step 1. Generate a new ρ_n from three previous ρ_1, ρ_2, ρ_3 by using the Quadratic Interpolation method.

Step 2. Fix $\rho = \rho_n$, perform the optimization based on SOCP algorithms. Update ρ_1, ρ_2, ρ_3.

M^4 : Extended into Regression--- Local Support Vector Regression (LSVR)

Regression: **Find a function** $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b, \quad \mathbf{w}, \mathbf{x} \in \mathbb{R}^d, b \in \mathbb{R}.$ **to approximate the data** $\{(\mathbf{x}_i, y_i) \mid \mathbf{x}_i \in \mathbb{R}^d,\ y_i \in \mathbb{R},\ i = 1, \ldots, N\}$

LSVR Model Definition

$$\begin{aligned}
\min_{\mathbf{w}, b, \xi_i, \xi_i^*} \quad & \frac{1}{N}\sum_{i=1}^{N}\sqrt{\mathbf{w}^T\Sigma_i\mathbf{w}} + C\sum_{i=1}^{N}(\xi_i + \xi_i^*), \\
\text{s.t.} \quad & y_i - (\mathbf{w}^T\mathbf{x}_i + b) \le \epsilon\sqrt{\mathbf{w}^T\Sigma_i\mathbf{w}} + \xi_i, \\
& (\mathbf{w}^T\mathbf{x}_i + b) - y_i \le \epsilon\sqrt{\mathbf{w}^T\Sigma_i\mathbf{w}} + \xi_i^*, \\
& \xi_i \ge 0, \quad \xi_i^* \ge 0, \quad i = 1, \ldots, N,
\end{aligned}$$

SVR Model Definition

$$\begin{aligned}
\min_{\mathbf{w}, b, \xi_i, \xi_i^*} \quad & \|\mathbf{w}\| + C\sum_{i=1}^{N}(\xi_i + \xi_i^*), \\
\text{s.t.} \quad & y_i - (\mathbf{w}\mathbf{x}_i + b) \le \epsilon + \xi_i, \\
& (\mathbf{w}\mathbf{x}_i + b) - y_i \le \epsilon + \xi_i^*, \\
& \xi_i \ge 0, \quad \xi_i^* \ge 0, \quad i = 1, \ldots, N.
\end{aligned}$$

$$\because y_i = w^T x_i + b, \quad \bar{y}_i = w^T \bar{x}_i + b$$

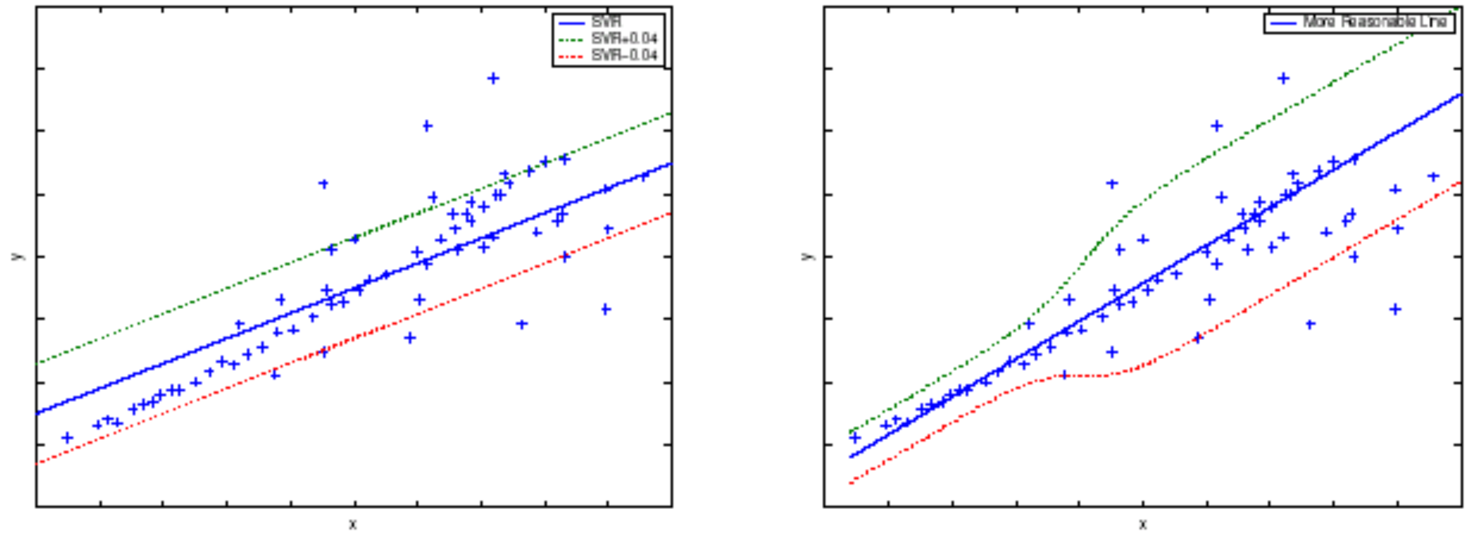
$$\because \Delta_i = \frac{1}{2k+1}\sum_{j=-k}^{k}(y_{i+j} - \bar{y}_i)^2 = \frac{1}{2k+1}\sum_{j=-k}^{k}[w^T(x_{i+j} - \bar{x}_i)]^2 = w^T\Sigma_i w$$

Local Support Vector Regression (LSVR)

- *When supposing Σ_i=I for each observation, LSVR is equivalent with l_1-SVR under a mild assumption.*

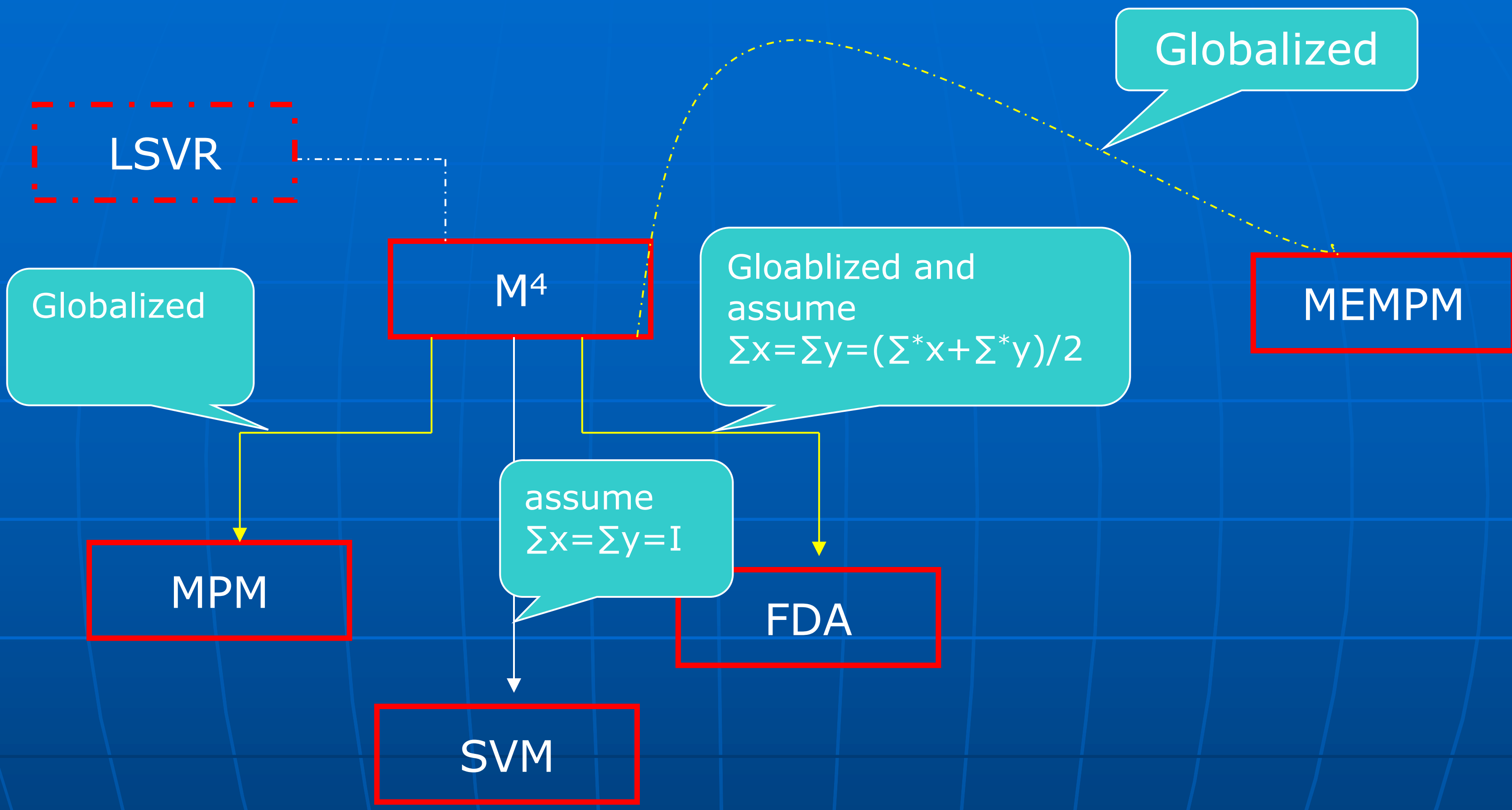
Lemma The LSVR model with setting $\Sigma_i = \mathbf{I}$ is equivalent to the ℓ_1-norm SVR in the sense that: (1) Assuming a unique ϵ_1^* exists for making ℓ_1-norm SVR optimal (i.e. setting ϵ to ϵ_1^* will make the objective function minimal), if for ϵ_1^* the ℓ_1-norm SVR achieves a solution $\{\mathbf{w}^*, b^*\} = \mathrm{SVR}(\epsilon_1^*)$, then the LSVR can produce the same solution by setting the parameter $\epsilon = \frac{\epsilon_1^*}{\|\mathbf{w}_1^*\|}$, i.e., $\mathrm{LSVR}(\frac{\epsilon_1^*}{\|\mathbf{w}_1^*\|}) = \mathrm{SVR}(\epsilon_1^*)$; (2) Assuming a unique ϵ_2^* exists for making the special case of LSVR optimal (i.e. setting ϵ to ϵ_2^* will make the objective function minimal), if for ϵ_2^* the special case of LSVR achieves a solution $\{\mathbf{w}_2^*, b_2^*\} = \mathrm{LSVR}(\epsilon_2^*)$, then the ℓ_1-norm SVR can produce the same solution by setting the parameter $\epsilon = \epsilon_2^*\|\mathbf{w}_2^*\|$, i.e., $\mathrm{SVR}(\epsilon_2^*\|\mathbf{w}_2^*\|) = \mathrm{LSVR}(\epsilon_1^*)$.

SVR vs. LSVR



(a) (b)

Short Summary



Non-linear Classifier : Kernelization (I)

- Previous discussions of MEMPM, BMPM, M^4 , and LSVR are conducted in the scope of linear classification.

- How about non-linear classification problems?

Using Kernelization techniques

Non-linear Classifier : Kernelization (II)

- In the next slides, we mainly discuss the kernelization on M^4, while the proposed kernelization method is also applicable for MEMPM, BMPM, and LSVR.

Nonlinear Classifier: Kernelization (III)

- Map data to higher dimensional feature space $\mathbf{R}^f$

$$\boxed{\begin{array}{l} \mathbf{x}_i \rightarrow \varphi(\mathbf{x}_i) \\ \mathbf{y}_i \rightarrow \varphi(\mathbf{y}_i) \end{array}}$$

- Construct the linear decision plane $\boxed{f(\gamma, b) = \gamma^T z + b \text{ in}}$ the feature space $\mathbf{R}^f$, **with** $\gamma \in \mathbf{R}^f$, $b \in \mathbf{R}$

- In $\mathbf{R}^f$, we need to solve

$$\max_{\rho, \boldsymbol{\gamma} \neq \mathbf{0}, b} \quad \rho \quad s.t.$$

$$\frac{(\boldsymbol{\gamma}^T \varphi(\mathbf{x}_i) + b)}{\sqrt{\boldsymbol{\gamma}^T \Sigma_{\varphi(\mathbf{x})} \boldsymbol{\gamma}}} \geq \rho, \quad i = 1, 2, \ldots, N_{\mathbf{x}} \ ,$$

$$\frac{-(\boldsymbol{\gamma}^T \varphi(\mathbf{y}_j) + b)}{\sqrt{\boldsymbol{\gamma}^T \Sigma_{\varphi(\mathbf{y})} \boldsymbol{\gamma}}} \geq \rho, \quad j = 1, 2, \ldots, N_{\mathbf{y}} \ .$$

- However, we do not want to solve this in an explicit form of φ. Instead, we want to solve it in a kernelization form

$$K(z_1, z_2) = \varphi(z_1)^T \varphi(z_2)$$

Nonlinear Classifier: Kernelization (IV)

Corollary If the estimates of means and covariance matrices are given in M^4 as the following plug-in estimates:

$$\overline{\varphi(\mathbf{x})} = \frac{1}{N_{\mathrm{x}}}\sum_{i=1}^{N_{\mathrm{x}}}\varphi(\mathbf{x}_i), \qquad \overline{\varphi(\mathbf{y})} = \frac{1}{N_{\mathrm{x}}}\sum_{j=1}^{N_{\mathrm{x}}}\varphi(\mathbf{y}_j),$$

$$\Sigma_{\varphi(\mathbf{x})} = \frac{1}{N_{\mathrm{y}}}\sum_{i=1}^{N_{\mathrm{x}}}(\varphi(\mathbf{x}_i) - \overline{\varphi(\mathbf{x})})(\varphi(\mathbf{x}_i) - \overline{\varphi(\mathbf{x})})^T,$$

$$\Sigma_{\varphi(\mathbf{y})} = \frac{1}{N_{\mathrm{y}}}\sum_{j=1}^{N_{\mathrm{y}}}(\varphi(\mathbf{y}_j) - \overline{\varphi(\mathbf{y})})(\varphi(\mathbf{y}_j) - \overline{\varphi(\mathbf{y})})^T,$$

then the optimal γ in (4.37-4.39) lies in the space spanned by the training points.

$$\gamma = \sum_{i=1}^{N_{\mathrm{x}}}\mu_i\varphi(\mathbf{x}_i) + \sum_{j=1}^{N_{\mathrm{y}}}\upsilon_j\varphi(\mathbf{y}_j),$$

Nonlinear Classifier: Kernelization (V)

Kernelization Theorem of M^4 *The optimal decision hyperplane for* M^4 *involves solving the following optimization problem:*

$$\max_{\rho,\boldsymbol{\eta}\neq\mathbf{0},b} \rho \quad s.t.$$

$$\frac{(\boldsymbol{\eta}^T\mathbf{K}_i+b)}{\sqrt{\frac{1}{N_{\mathrm{x}}}\boldsymbol{\eta}^T\tilde{\mathbf{K}}_{\mathrm{x}}^T\tilde{\mathbf{K}}_{\mathrm{x}}\boldsymbol{\eta}}} \geq \rho, \quad i=1,2,\ldots,N_{\mathrm{x}}\ ,$$

$$\frac{-(\boldsymbol{\eta}^T\mathbf{K}_{j+N_{\mathrm{x}}}+b)}{\sqrt{\frac{1}{N_{\mathrm{y}}}\boldsymbol{\eta}^T\tilde{\mathbf{K}}_{\mathrm{y}}^T\tilde{\mathbf{K}}_{\mathrm{y}}\boldsymbol{\eta}}} \geq \rho, \quad j=1,2,\ldots,N_{\mathrm{y}}\ .$$

Notation

$$\boldsymbol{\eta} := [\mu_1,\ldots,\mu_{N_{\mathbf{x}}},\upsilon_1,\ldots,\upsilon_{N_{\mathbf{y}}}]^T$$

$$\tilde{\mathbf{k}}_{\mathbf{x}},\tilde{\mathbf{k}}_{\mathbf{y}} \in \mathbb{R}^{N_{\mathbf{x}}+N_{\mathbf{y}}} \qquad [\tilde{\mathbf{k}}_{\mathbf{x}}]_i := \frac{1}{N_{\mathbf{x}}}\sum_{j=1}^{N_{\mathbf{x}}}\mathbf{K}(\mathbf{x}_j,\mathbf{z}_i)\ .$$

$$[\tilde{\mathbf{k}}_{\mathbf{y}}]_i := \frac{1}{N_{\mathbf{y}}}\sum_{j=1}^{N_{\mathbf{y}}}\mathbf{K}(\mathbf{y}_j,\mathbf{z}_i)\ .$$

$$\mathbf{1}_{N_{\mathbf{x}}} \in \mathbb{R}^{N_{\mathbf{x}}} \qquad \mathbf{1}_i := 1 \quad i=1,2,\ldots N_{\mathbf{x}}\ .$$

$$\mathbf{1}_{N_{\mathbf{y}}} \in \mathbb{R}^{N_{\mathbf{y}}} \qquad \mathbf{1}_i := 1 \quad i=1,2,\ldots N_{\mathbf{y}}\ .$$

$$\tilde{\mathbf{K}} := \begin{pmatrix}\tilde{\mathbf{K}}_{\mathbf{x}}\\ \tilde{\mathbf{K}}_{\mathbf{y}}\end{pmatrix} := \begin{pmatrix}\mathbf{K}_{\mathbf{x}}-\mathbf{1}_{N_{\mathbf{x}}}\tilde{\mathbf{k}}_{\mathbf{x}}^T\\ \mathbf{K}_{\mathbf{y}}-\mathbf{1}_{N_{\mathbf{y}}}\tilde{\mathbf{k}}_{\mathbf{y}}^T\end{pmatrix}\ .$$

Experimental Results ---MEMPM (I)

Six benchmark data sets From UCI Repository

	Dataset	Attributes #	Instances#
1	Twonorm	20	7400
2	Breast	9	699
3	Ionosphere	34	351
4	Pima	8	768
5	Heart-disease	13	270
6	Vote	16	435

Platform: Windows 2000
Developing tool: Matlab 6.5

Evaluate both the linear and the Gaussian kernel with the wide parameter for Gaussian chosen by cross validations.

Experimental Results ---MEMPM(II)

Dataset	MEMPM				MPM	
	α	β	$\theta\alpha + (1-\theta)\beta$	Accuracy	α	Accuracy
Twonorm(%)	$80.3 \pm 0.2\%$	$79.9 \pm 0.1\%$	$80.1 \pm 0.1\%$	$\boxed{97.9 \pm 0.1\%}$	$80.1 \pm 0.1\%$	$\boxed{97.9 \pm 0.1\%}$
Breast(%)	$77.8 \pm 0.8\%$	$91.4 \pm 0.5\%$	$86.7 \pm 0.5\%$	$96.9 \pm 0.3\%$	$84.4 \pm 0.5\%$	$\boxed{97.0 \pm 0.2\%}$
Ionosphere(%)	$95.9 \pm 1.2\%$	$36.5 \pm 2.6\%$	$74.5 \pm 0.8\%$	$\boxed{88.5 \pm 1.0\%}$	$63.4 \pm 1.1\%$	$84.8 \pm 0.8\%$
Pima(%)	$0.9 \pm 0.0\%$	$62.9 \pm 1.1\%$	$41.3 \pm 0.8\%$	$\boxed{76.8 \pm 0.6\%}$	$32.0 \pm 0.8\%$	$76.1 \pm 0.6\%$
Heart-disease(%)	$43.6 \pm 2.5\%$	$66.5 \pm 1.5\%$	$56.3 \pm 1.4\%$	$\boxed{84.2 \pm 0.7\%}$	$54.9 \pm 1.4\%$	$83.2 \pm 0.8\%$
Vote(%)	$82.6 \pm 1.3\%$	$84.6 \pm 0.7\%$	$83.9 \pm 0.9\%$	$\boxed{94.9 \pm 0.4\%}$	$83.8 \pm 0.9\%$	$94.8 \pm 0.4\%$

Table 2: Lower bound α, β, and test accuracy compared to MPM in the linear setting.

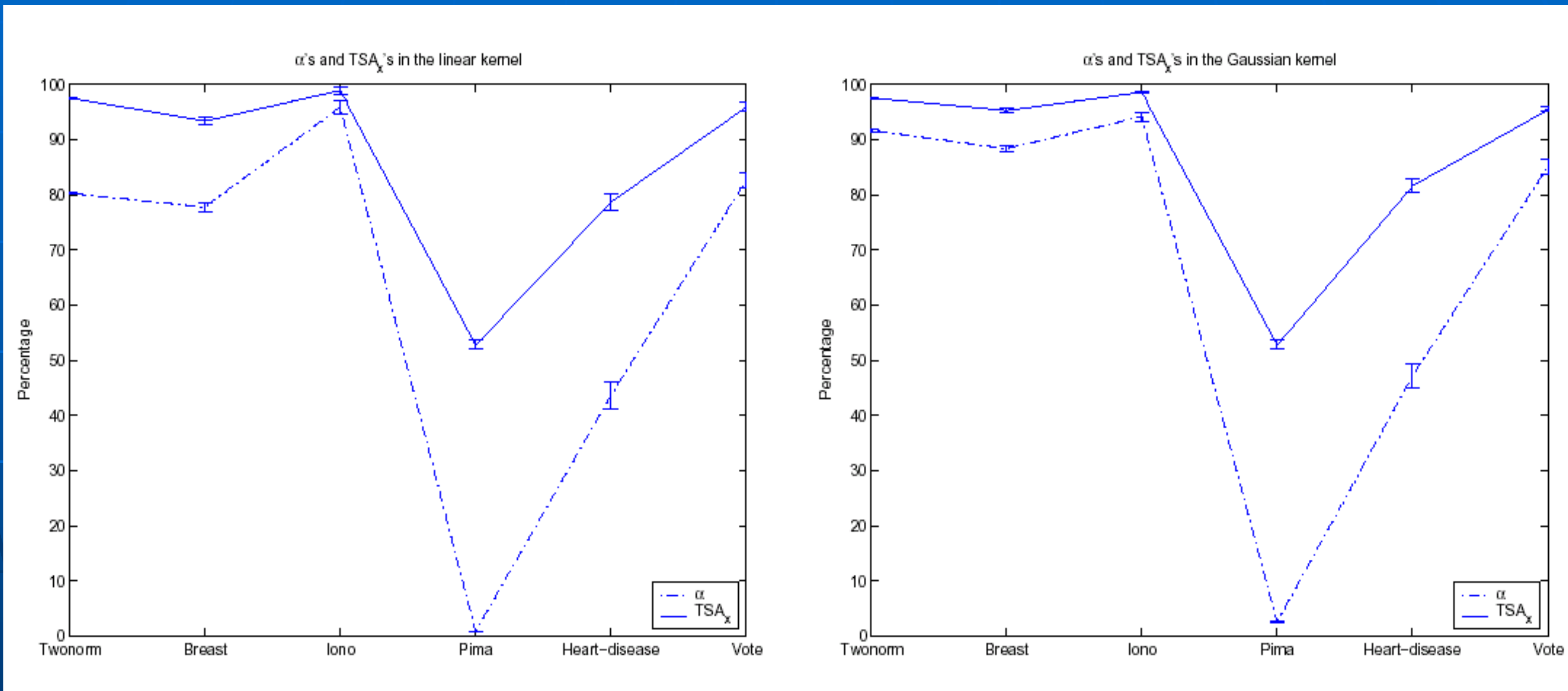
Dataset	MEMPM				MPM	
	α	β	$\theta\alpha + (1-\theta)\beta$	Accuracy	α	Accuracy
Twonorm(%)	$91.7 \pm 0.2\%$	$91.7 \pm 0.2\%$	$91.7 \pm 0.2\%$	$\boxed{97.9 \pm 0.1\%}$	$91.7 \pm 0.2\%$	$\boxed{97.9 \pm 0.1\%}$
Breast(%)	$88.4 \pm 0.6\%$	$90.7 \pm 0.4\%$	$89.9 \pm 0.4\%$	$\boxed{96.9 \pm 0.2\%}$	$89.9 \pm 0.4\%$	$\boxed{96.9 \pm 0.3\%}$
Ionosphere(%)	$94.2 \pm 0.8\%$	$80.9 \pm 3.0\%$	$89.4 \pm 0.8\%$	$\boxed{93.8 \pm 0.4\%}$	$89.0 \pm 0.8\%$	$92.2 \pm 0.4\%$
Pima(%)	$2.6 \pm 0.1\%$	$62.3 \pm 1.6\%$	$41.4 \pm 1.1\%$	$\boxed{77.0 \pm 0.7\%}$	$32.1 \pm 1.0\%$	$76.2 \pm 0.6\%$
Heart-disease(%)	$47.1 \pm 2.2\%$	$66.6 \pm 1.4\%$	$58.0 \pm 1.5\%$	$\boxed{83.9 \pm 0.9\%}$	$57.4 \pm 1.6\%$	$83.1 \pm 1.0\%$
Vote(%)	$85.1 \pm 1.3\%$	$84.3 \pm 0.7\%$	$84.7 \pm 0.8\%$	$\boxed{94.7 \pm 0.5\%}$	$84.4 \pm 0.8\%$	$94.6 \pm 0.4\%$

Table 3: Lower bound α, β, and test accuracy compared to MPM with the Gaussian kernel.

At the Significance level 0.05

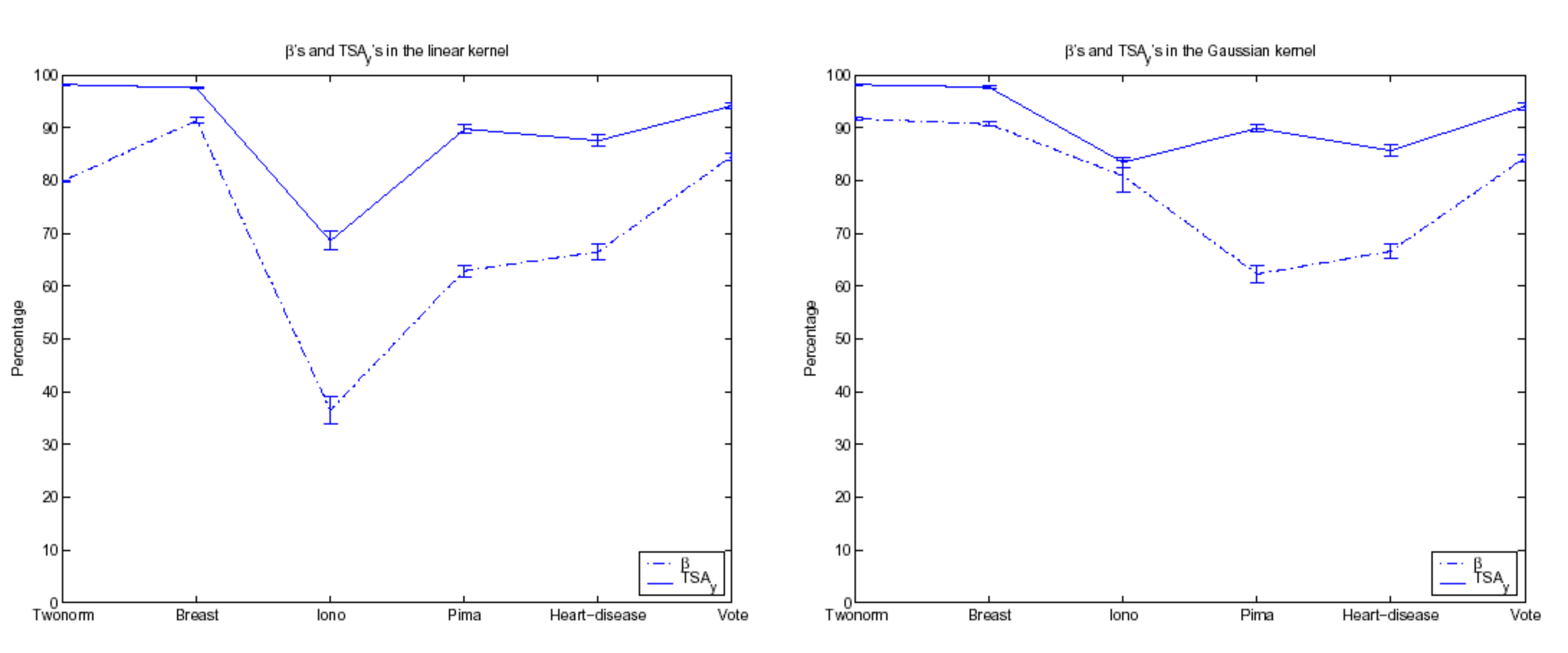
Experimental Results ---MEMPM (III)

α vs. The test-set accuracy for x (TSAx)



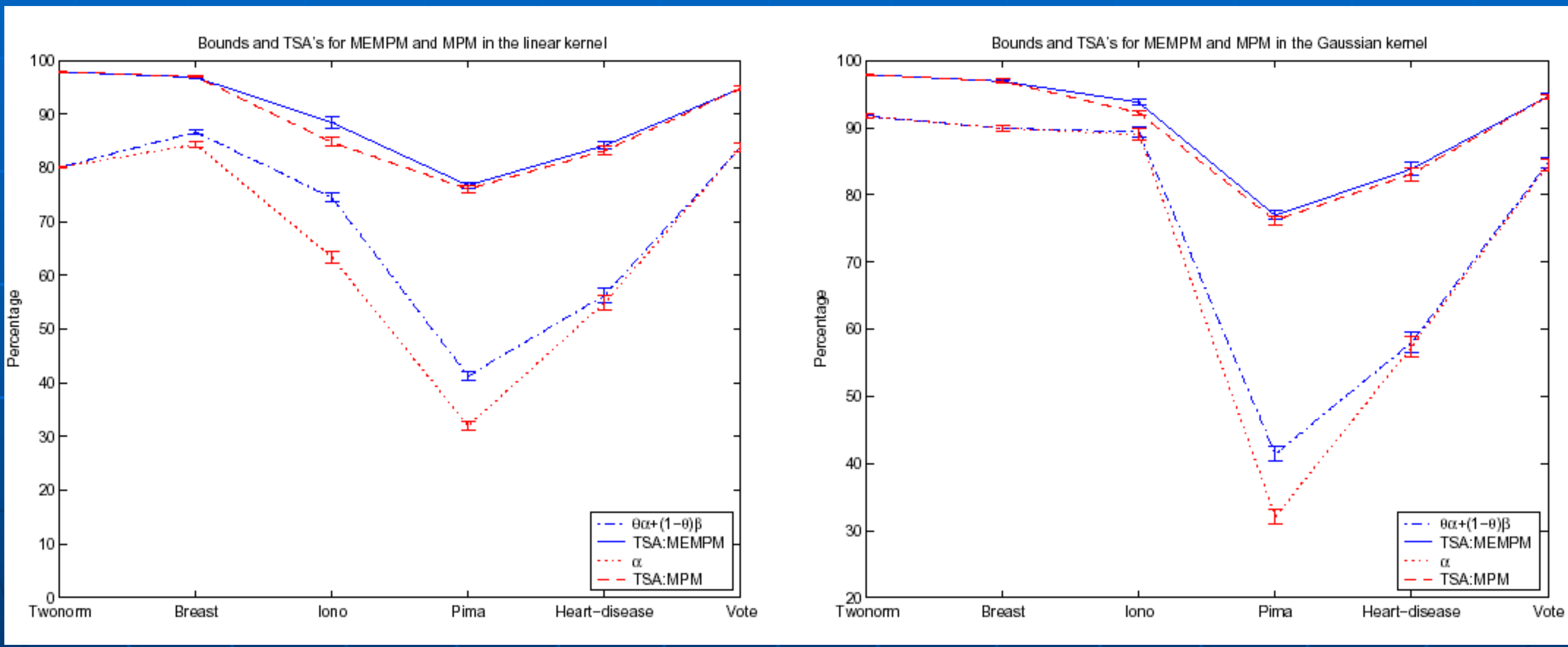
Experimental Results ---MEMPM (IV)

β *vs. The test-set accuracy for y (TSAy)*



Experimental Results ---MEMPM (V)

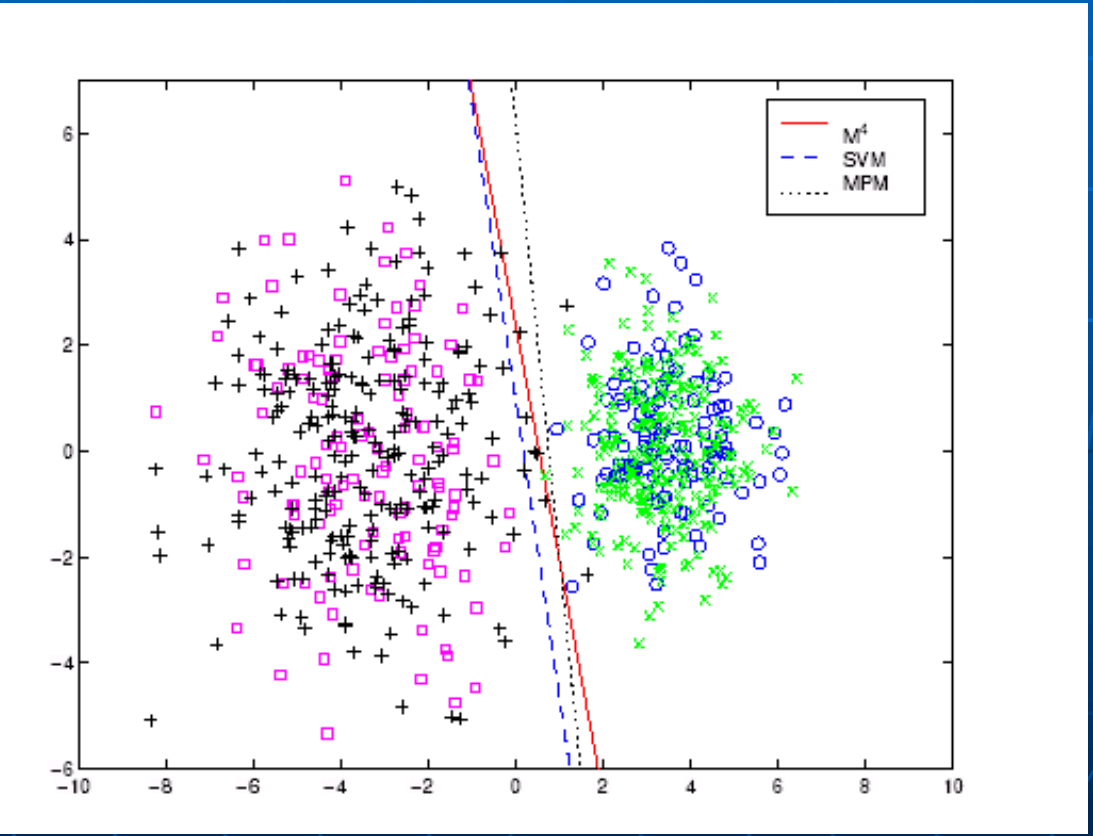
$\theta\alpha + (1-\theta)\beta$ *vs. The overall test-set accuracy (TSA)*



Experimental Results ---M^4 (I)

- Synthetic Toy Data (1)

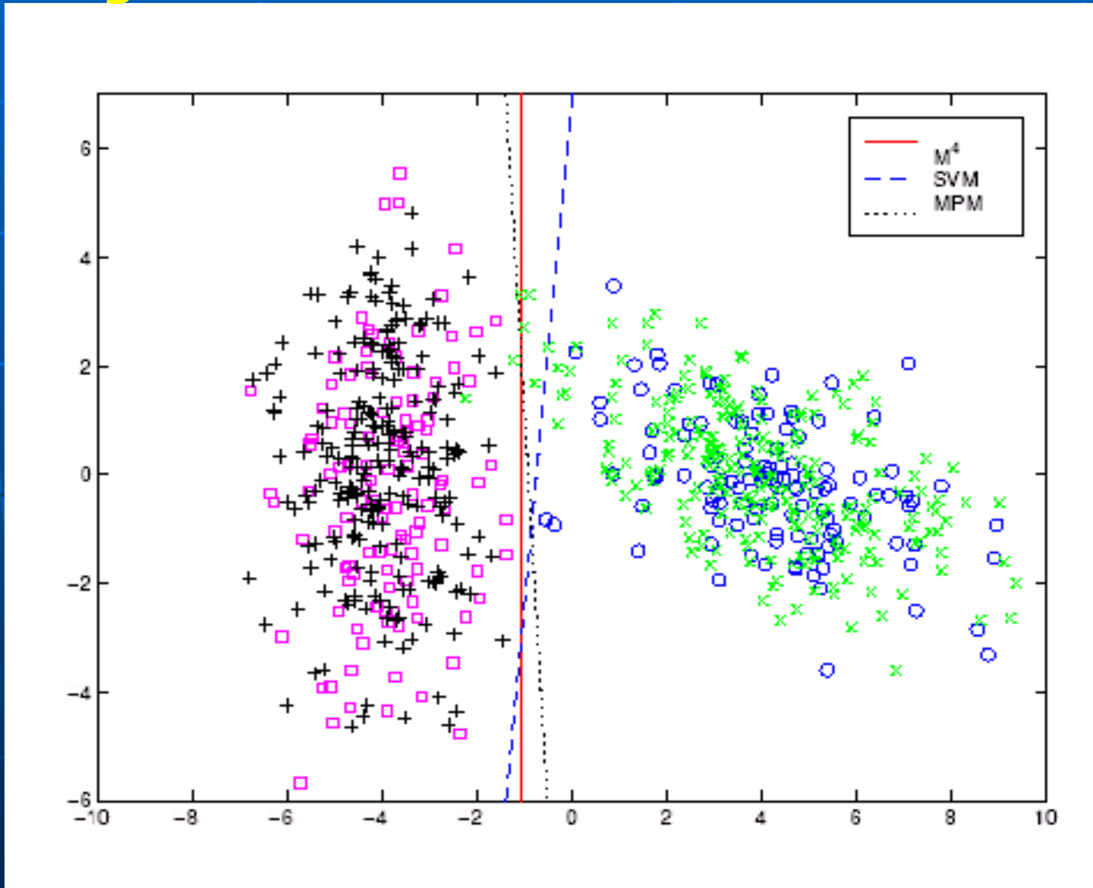
*Two types of data with **the same data orientation** but **different data magnitude***



Experimental Results ---M^4 (II)

- Synthetic Toy Data (2)

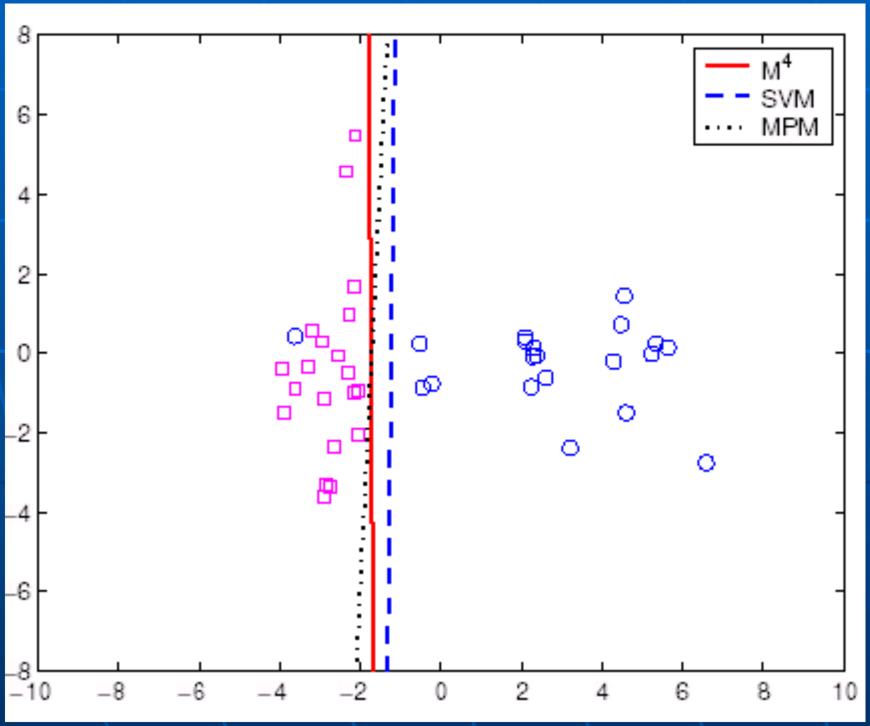
Two types of data with ***the same data magnitude*** *but* ***different data orientation***

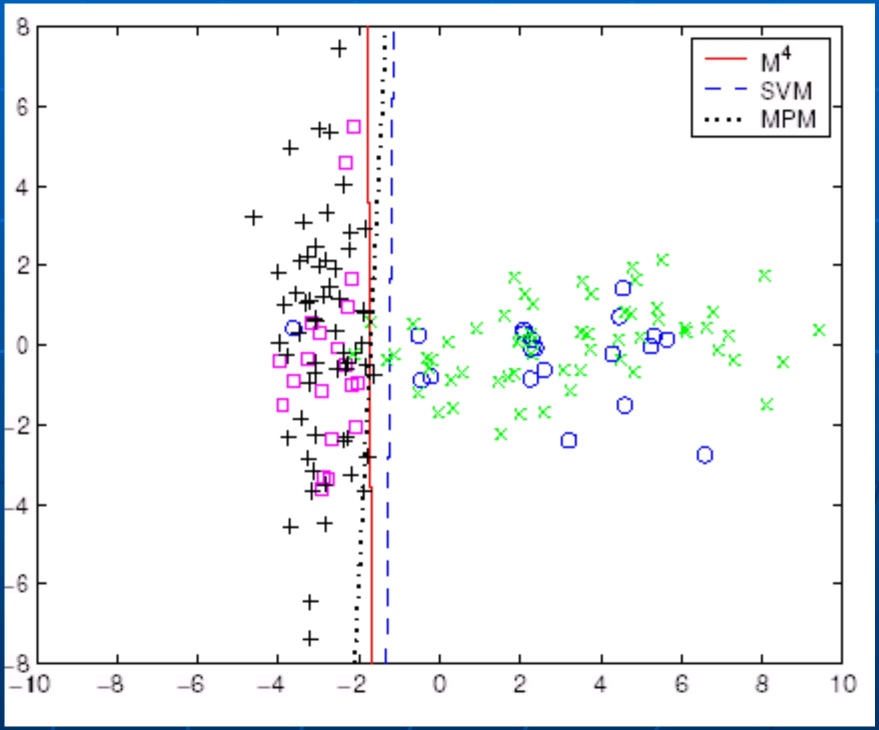


Experimental Results ---M^4 (III)

- Synthetic Toy Data (3)

Two types of data with ***the different data magnitude*** *and* ***different data orientation***





Experimental Results ---M^4 (IV)

- Benchmark Data from UCI

Data set	Linear kernel			Gaussian kernel		
	M^4	SVM	MPM	M^4	SVM	MPM
Twonorm(%)	96.5 ± 0.6	95.1 ± 0.7	$\mathbf{97.6 \pm 0.5}$	96.5 ± 0.7	96.1 ± 0.4	$\mathbf{97.6 \pm 0.5}$
Breast(%)	$\mathbf{97.5 \pm 0.7}$	96.6 ± 0.5	96.9 ± 0.8	$\mathbf{97.5 \pm 0.6}$	96.7 ± 0.4	96.9 ± 0.8
Ionosphere(%)	$\mathbf{87.7 \pm 0.8}$	86.9 ± 0.6	84.8 ± 0.8	$\mathbf{94.5 \pm 0.4}$	94.2 ± 0.3	92.3 ± 0.6
Pima(%)	77.7 ± 0.9	$\mathbf{77.9 \pm 0.7}$	76.1 ± 1.2	77.6 ± 0.8	$\mathbf{78.0 \pm 0.5}$	76.2 ± 1.2
Sonar(%)	$\mathbf{77.6 \pm 1.2}$	76.2 ± 1.1	75.5 ± 1.1	84.9 ± 1.2	86.5 ± 1.1	$\mathbf{87.3 \pm 0.8}$
Vote(%)	$\mathbf{96.1 \pm 0.5}$	95.1 ± 0.4	94.8 ± 0.4	$\mathbf{96.2 \pm 0.5}$	95.9 ± 0.6	94.6 ± 0.4
Heart-disease(%)	$\mathbf{86.6 \pm 0.8}$	84.1 ± 0.7	83.2 ± 0.8	$\mathbf{86.2 \pm 0.8}$	83.8 ± 0.5	83.1 ± 1.0

Table 2: Comparisons of classification accuracies among M^4, SVM, and MPM.

Future Work

- Speeding up M^4 and MEMPM
  - Contain support vectors—can we employ its sparsity as has been done in SVM?
  - Can we reduce redundant points??
- How to impose constrains on the kernelization for keeping the topology of data?
- Generalization error bound?
  - SVM and MPM have both error bounds.
- How to extend to multi-category classifications?
  - One vs. One or One vs. All?
  - Or seeking a principled way to construct multi-way boundary **in a step**??

Conclusion

- We propose a general global learning model MEMPM
  - A Worst-case distribution-free Bayes Optimal classifier
  - Containing an explicit error bound for future data
  - Subsuming BMPM which is idea for biased classification
- We propose a hybrid framework M^4 by learning from data locally and globally
  - This model subsumes three important models as special cases
    - SVM
    - MPM
    - FDA
  - Extended into regression tasks

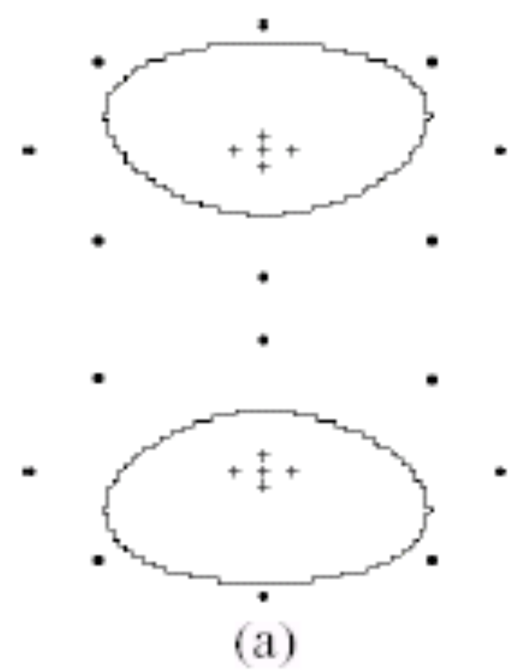
Discussion (I)

- In **linear** cases, M^4 **outperforms SVM and MPM**

- In **Gaussian** cases, M^4 is **slightly better or comparable than SVM**
  - (1) Sparsity in the feature space results in inaccurate estimation of covariance matrices
  - (2) Kernelization may not keep data topology of the original data.— Maximizing Margin in the feature space does not necessarily maximize margin in the original space

Discussion (II)

An example to illustrate that maximizing the margin in the feature space does not necessarily maximize the margin in the original space



(a) SVM using degree 4 polynomial kernel.

From Simon Tong et al. *Restricted Bayesian Optimal classifiers*, AAAI, 2000.

Setup

- Three concerns:
  - Binary classification data sets
  - For easy comparison. MPM *(Lanckriet et al. JMLR 02 or nips02)* also uses these data sets.
  - Medium or smaller size Data sets

Appendix A: MEMPM- BMPM (I)

1

Lemma Given $\mathbf{w} \neq \mathbf{0}$ and b, such that $\mathbf{w}^T\mathbf{y} \leq b$ and $\beta \in [0,1)$, the condition

$$\inf_{\mathbf{y}\sim(\overline{\mathbf{y}},\Sigma_{\mathbf{y}})} \mathbf{Pr}\{\mathbf{w}^T\mathbf{y} \leq b\} \geq \beta,$$

holds if and only if $b - \mathbf{w}^T\overline{\mathbf{y}} \geq \kappa(\beta)\sqrt{\mathbf{w}^T\Sigma_{\mathbf{y}}\mathbf{w}}$ with $\kappa(\beta) = \sqrt{\frac{\beta}{1-\beta}}$.

2

$$\max_{\alpha,\mathbf{w}\neq\mathbf{0},b} \quad \alpha \quad \text{s.t.}$$

$$-b + \mathbf{w}^T\overline{\mathbf{x}} \geq \kappa(\alpha)\sqrt{\mathbf{w}^T\Sigma_{\mathbf{x}}\mathbf{w}}\ ,$$

$$b - \mathbf{w}^T\overline{\mathbf{y}} \geq \kappa(\beta_0)\sqrt{\mathbf{w}^T\Sigma_{\mathbf{y}}\mathbf{w}}\ ,$$

3

$$\mathbf{w}^T\overline{\mathbf{y}} + \kappa(\beta_0)\sqrt{\mathbf{w}^T\Sigma_{\mathbf{y}}\mathbf{w}} \leq b \leq \mathbf{w}^T\overline{\mathbf{x}} - \kappa(\alpha)\sqrt{\mathbf{w}^T\Sigma_{\mathbf{x}}\mathbf{w}}\ .$$

4

$$\max_{\alpha,\mathbf{w}\neq\mathbf{0}} \quad \alpha \quad \text{s.t.}$$

$$1 \geq \kappa(\alpha)\sqrt{\mathbf{w}^T\Sigma_{\mathbf{x}}\mathbf{w}} + \kappa(\beta_0)\sqrt{\mathbf{w}^T\Sigma_{\mathbf{y}}\mathbf{w}}\ ,$$

$$\mathbf{w}^T(\overline{\mathbf{x}} - \overline{\mathbf{y}}) = 1.$$

5

$$\max_{\mathbf{w}\neq\mathbf{0}} \frac{1 - \kappa(\beta_0)\sqrt{\mathbf{w}^T\Sigma_{\mathbf{y}}\mathbf{w}}}{\sqrt{\mathbf{w}^T\Sigma_{\mathbf{x}}\mathbf{w}}} \quad \text{s.t.} \quad \mathbf{w}^T(\overline{\mathbf{x}} - \overline{\mathbf{y}}) = 1.$$

Fractional Programming

Appendix A: MEMPM- BMPM (II)

Solving Fractional Programming problem

- Parametric Method

  Find $\mathbf{a}$ by solving

$$\max_{\mathbf{a}\neq\mathbf{0}} \quad 1-\kappa(\gamma)\sqrt{\mathbf{a}^T\Sigma_{\mathbf{y}}\mathbf{a}}-\lambda\sqrt{\mathbf{a}^T\Sigma_{\mathbf{x}}\mathbf{a}} \quad \text{s.t.} \quad \mathbf{a}^T(\overline{\mathbf{x}}-\overline{\mathbf{y}})=1$$

  Update

$$\lambda \leftarrow \frac{1-\kappa(\gamma)\sqrt{\mathbf{a}^T\Sigma_{\mathbf{y}}\mathbf{a}}}{\sqrt{\mathbf{a}^T\Sigma_{\mathbf{x}}\mathbf{a}}}$$

- Equivalently

$$\min_{\mathbf{a}\neq\mathbf{0}} \quad \kappa(\gamma)\sqrt{\mathbf{a}^T\Sigma_{\mathbf{y}}\mathbf{a}}+\lambda\sqrt{\mathbf{a}^T\Sigma_{\mathbf{x}}\mathbf{a}} \quad \text{s.t.} \quad \mathbf{a}^T(\overline{\mathbf{x}}-\overline{\mathbf{y}})=1$$

- Least-squares approach

Appendix B: Optimization of LSVR(I)

$$
\begin{aligned}
\min_{\mathbf{w},b,t_i,\xi_i,\xi_i^*} \quad & \frac{1}{N}\sum_{i=1}^{N} t_i + C\sum_{i=1}^{N}(\xi_i + \xi_i^*), \\
\text{s.t.} \quad & y_i - (\mathbf{w}^T\mathbf{x}_i + b) \le \epsilon\sqrt{\mathbf{w}^T\Sigma_i\mathbf{w}} + \xi_i, \\
& (\mathbf{w}^T\mathbf{x}_i + b) - y_i \le \epsilon\sqrt{\mathbf{w}^T\Sigma_i\mathbf{w}} + \xi_i^*, \\
& \sqrt{\mathbf{w}^T\Sigma_i\mathbf{w}} \le t_i,
\end{aligned}
$$

$\Downarrow$

Hard to be solved…

Appendix B: Optimization of LSVR(II)

Can be relaxed as the following:

$$\min_{\mathbf{w},b,t_i,\xi_i,\xi_i^*} \quad \frac{1}{N}\sum_{i=1}^{N} t_i + C\sum_{i=1}^{N}(\xi_i + \xi_i^*),$$

$$\text{s.t.} \quad y_i - (\mathbf{w}^T\mathbf{x}_i + b) \leq \epsilon t_i + \xi_i,$$

$$(\mathbf{w}^T\mathbf{x}_i + b) - y_i \leq \epsilon t_i + \xi_i^*,$$

$$\sqrt{\mathbf{w}^T\Sigma_i\mathbf{w}} \leq t_i,$$

$$t_i \geq 0, \xi_i \geq 0, \xi_i^* \geq 0, i = 1, \ldots, N.$$

Second-Order Cone Programming

Appendix C: Convex Optimization

Linear Program:

$$\min_x c^T x \quad \text{s.t.} \quad Ax \preceq b$$
$$Fx = g$$

LP SVM
(Mangasarian, Bennett)

Quadratic Program:

$$\min_x x^T P x + 2q^T x + r \quad \text{s.t.} \quad Ax \preceq b$$
$$Fx = g$$

SVM
(Vapnik)

Quadratic Constrained Quadratic Program:

$$\min_x x^T P_0 x + 2q_0^T x + r_0 \quad \text{s.t.} \quad x^T P_i x + 2q_i^T x + r_i \leq 0$$
$$i = 1, \ldots, L \quad (\text{convex if} P_i \succeq 0)$$

Kernel Fisher Discriminant
(Mika et al.)

Second Order Cone Program:

$$\min_x c^T x \quad \text{s.t.} \quad \|A_i x + b_i\|_2 \leq e_i^T x + d_i$$
$$i = 1, \ldots, L$$

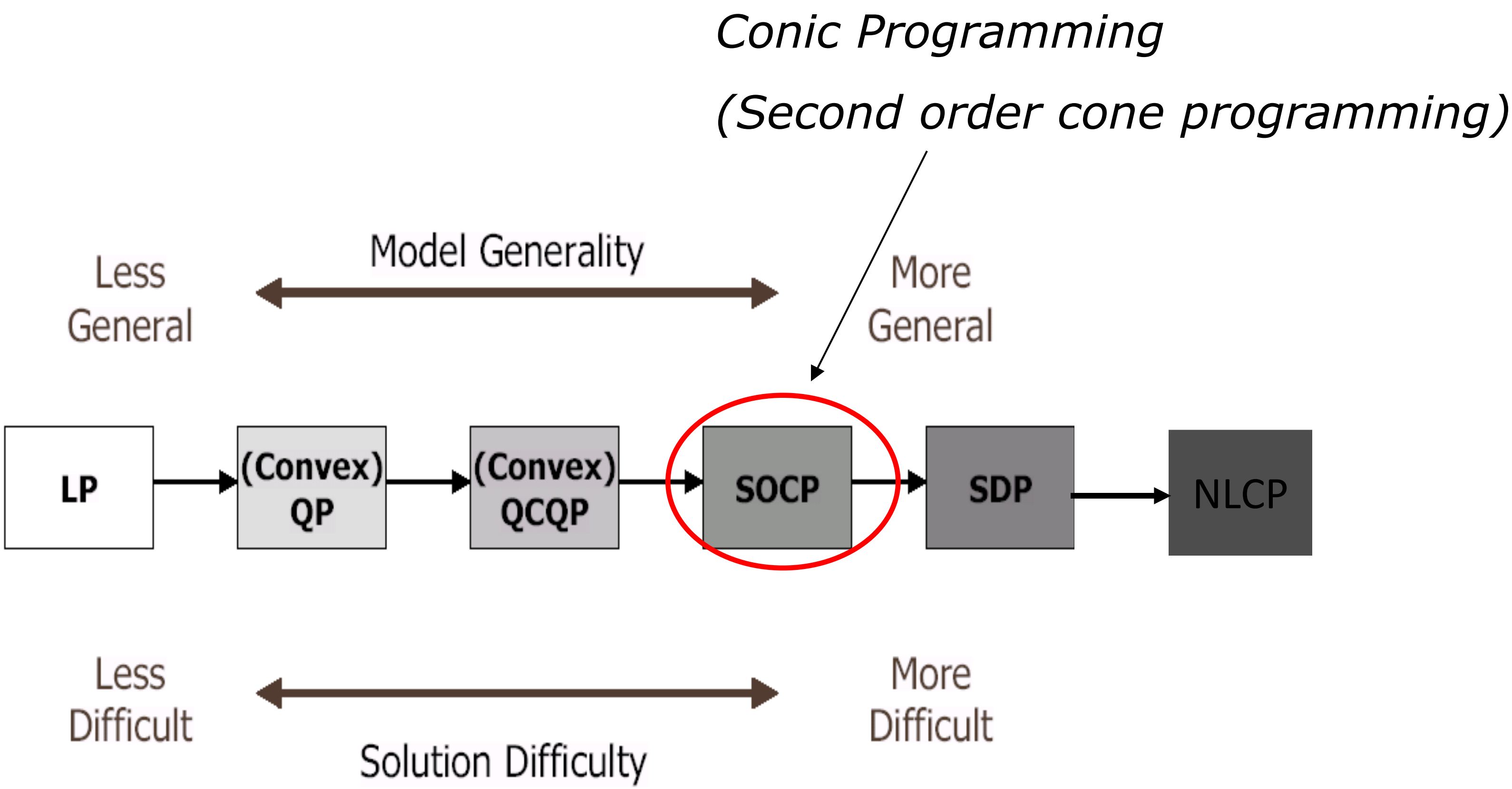
Minimax Probability Machine
(Lanckriet et al.)

Semi-Definite Program:

$$\min_x c^T x \quad \text{s.t.} \quad A(x) = A_0 + \sum_{i=1}^{n} x_i A_i \succeq 0 \ \ (A_i = A_i^T \in \mathbb{R}^{p\times p})$$
$$Fx = g$$

Kernel matrix learning
(Lanckriet, Cristianini et al.)

Appendix C: Convex Optimization



Appendix C: Convex Optimization - SOCP

$$\min \quad f^T x$$
$$s.t. \quad \|C_i x + d_i\| \le a_i^T x + b_i, \qquad i = 1,\dots, N$$

$$x \in \Re^n \quad f, a_i \in \Re^n \quad C_i \in \Re^{(k_i-1)\times n} \quad d_i \in \Re^{k_i-1} \quad b_i \in \Re$$

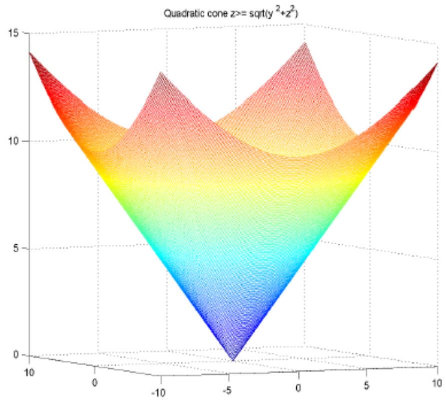
Equivalent to conic program

- Linear constraints: cone dimension k=1
- Cone constraints: change of variables

$$(\text{vector})\ y = C_i x + d_i, \qquad z = a_i^T x + b_i$$

8

Quadractic cone C sometimes also called Lorentz cone (or ice cream cone)



Trivial Quadratic Cone:

$$z \ge \sqrt{x^2 + y^2}$$

7

Appendix C: SOCP-Solver

Sedumi (MATLAB)

Loqo (C, MATLAB)

MOSEK (C, MATLAB)

SDPT3 (MATLAB+C or FORTRAN)

The worst-case cost is $O(n^3)$

Time Complexity

Models	Time Complexity
MEMPM	$O(Ln^3+Nn^2)$
BMPM	$O(n^3+Nn^2)$
M^4	$O(Nn^3)$
LS-SVM	$O(n^3+Nn^2)$
LSVR	$O(Nn^3)$
LS-SVR	$O(n^3+Nn^2)$

Time Complexity

Thus we believe that for practical purposes the cost of solving an SOCP is roughly equal to the cost of solving a modest number (5–50) of systems of the form (40). If no special structure in the problem data is exploited, the cost of solving the system is $O(n^3)$, and the cost of forming the system matrix is $O(n^2 \sum_{i=1}^{N} n_i)$. In practice, special problem structure (*e.g.*, sparsity) often allows forming the equations faster, or solving the systems (39) or (40) more efficiently.

----"Applications of Second Order Cone Programming", Lobo, Boyd et al. in Linear Algebra and Applications.