M L
D M

Chapter 9
Support Vector Machines

Learning Machines

- A machine to learn the mapping

$$\mathbf{x}_i \text{ a } y_i$$

- Defined as

$$\mathbf{x} \text{ a } f(\mathbf{x}, \underbrace{\boldsymbol{\alpha}}_{})$$

Learning by adjusting this parameter?

Generalization vs. Learning

- How a machine learns?
  - Adjusting the parameters so as to partition the pattern (feature) space for classification.
  - How to adjust?

    Minimize the empirical risk (traditional approaches).

- What the machine learned?
  - Memorize the patterns it sees? or
  - Memorize the rules it finds for different classes?
  - What does the machine actually learn if it minimizes empirical risk only?

Risks

Expected Risk (test error)

$$R(\boldsymbol{\alpha}) = \int \tfrac{1}{2}\left| y - f(\mathbf{x}, \boldsymbol{\alpha}) \right| dP(\mathbf{x}, y)$$

Empirical Risk (training error)

$$R_{emp}(\boldsymbol{\alpha}) = \tfrac{1}{2l}\sum_{i=1}^{l}\left| y_i - f(\mathbf{x}_i, \boldsymbol{\alpha}) \right|$$

$$R(\boldsymbol{\alpha}) \approx R_{emp}(\boldsymbol{\alpha})?$$

More on Empirical Risk

- How can make the empirical risk arbitrarily small?
  - To let the machine have very large memorization capacity.
- Does a machine with small empirical risk also get small expected risk?
- How to avoid the machine to strain to memorize training patterns, instead of doing generalization, only?
- How to deal with the straining-memorization capacity of a machine?
- What the new criterion should be?

Structure Risk Minimization

Goal: Learn both the right ‘structure’ and right `rules’ for classification.

Right Structure:

E.g., Right amount and right forms of components or parameters are to participate in a learning machine.

Right Rules:

The empirical risk will also be reduced if right rules are learned.

New Criterion

Total Risk = Empirical Risk + Risk due to the structure of the learning machine

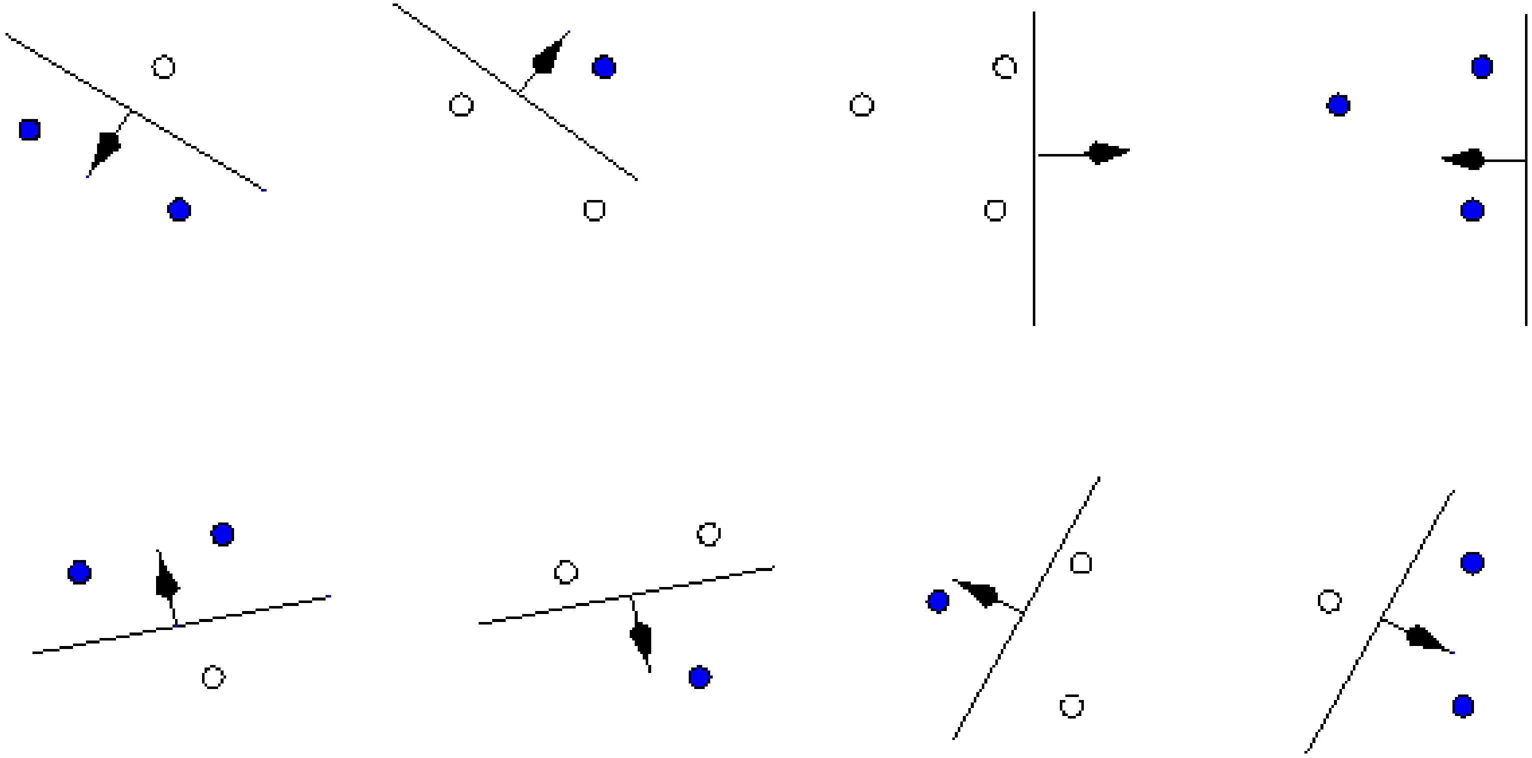
The VC Dimension

- Consider a set of function $f(\mathbf{x}, \alpha) \in \{-1,1\}$.
- A given set of l points can be labeled in 2^l ways.
- If a member of the set $\{f(\alpha)\}$ can be found which correctly assigns the labels for all labeling, then the set of points is *shattered* by that set of functions.
- The *VC dimension* of $\{f(\alpha)\}$ is the maximum number of training points that can be shattered by $\{f(\alpha)\}$.

VC: Vapnik Chervonenkis

The VC Dimension for Oriented Lines in R^2

- VC dimension = 3



More on VC Dimension

- In general, the VC dimension of a set of oriented hyperplanes in R^n is $n+1$.
- VC dimension is a measure of memorization capability.
- VC dimension is *not* directly related to number of parameters. Vapnik (1995) has an example with 1 parameter and infinite VC dimension.

Bound on Expected Risk

Expected Risk $$R(\boldsymbol{\alpha}) = \int \tfrac{1}{2} \left| y - f(\mathbf{x}, \boldsymbol{\alpha}) \right| dP(\mathbf{x}, y)$$

Empirical Risk $$R_{emp}(\boldsymbol{\alpha}) = \tfrac{1}{2l} \sum_{i=1}^{l} \left| y_i - f(\mathbf{x}_i, \boldsymbol{\alpha}) \right|$$

$$P\left(R(\alpha) \le R_{emp}(\alpha) + \underbrace{\sqrt{\frac{h(\log(2l/h)+1) - \log(\eta/4)}{l}}}_{\text{VC Confidence}} \right) = 1 - \eta$$

h is the VC dimension; l is the number of samples

Bound on Expected Risk

Consider small η (e.g., $\eta \le 0.05$).

$$\Rightarrow \quad R(\alpha) \le R_{emp}(\alpha) + \sqrt{\frac{h(\log(2l/h)+1)-\log(\eta/4)}{l}}$$

$$P\left(R(\alpha) \le R_{emp}(\alpha) + \underbrace{\sqrt{\frac{h(\log(2l/h)+1)-\log(\eta/4)}{l}}}_{\text{VC Confidence}} \right) = 1-\eta$$

Bound on Expected Risk

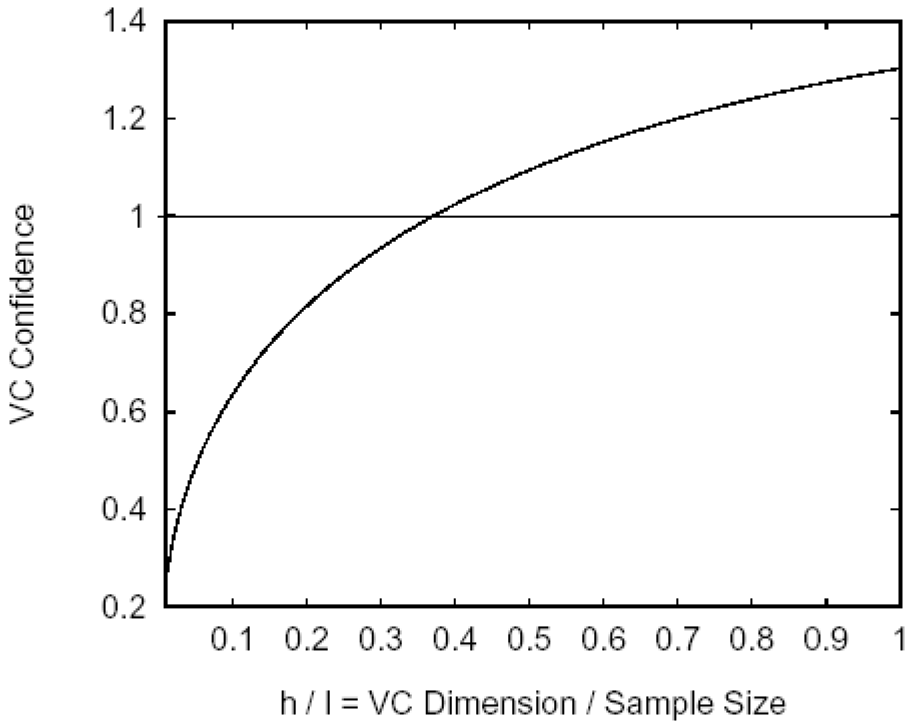
Consider small η (e.g., $\eta \leq 0.05$).

$$\Rightarrow \quad R(\alpha) \leq \underbrace{\underbrace{R_{emp}(\alpha)}_{\text{Traditional approaches minimize empirical risk only}} + \sqrt{\frac{h(\log(2l/h)+1)-\log(\eta/4)}{l}}}_{\text{Structure risk minimization want to minimize the bound}}$$

VC Confidence

MIMA

$$R(\alpha) \le R_{emp}(\alpha) + \sqrt{\frac{h(\log(2l/h)+1) - \log(\eta/4)}{l}}$$



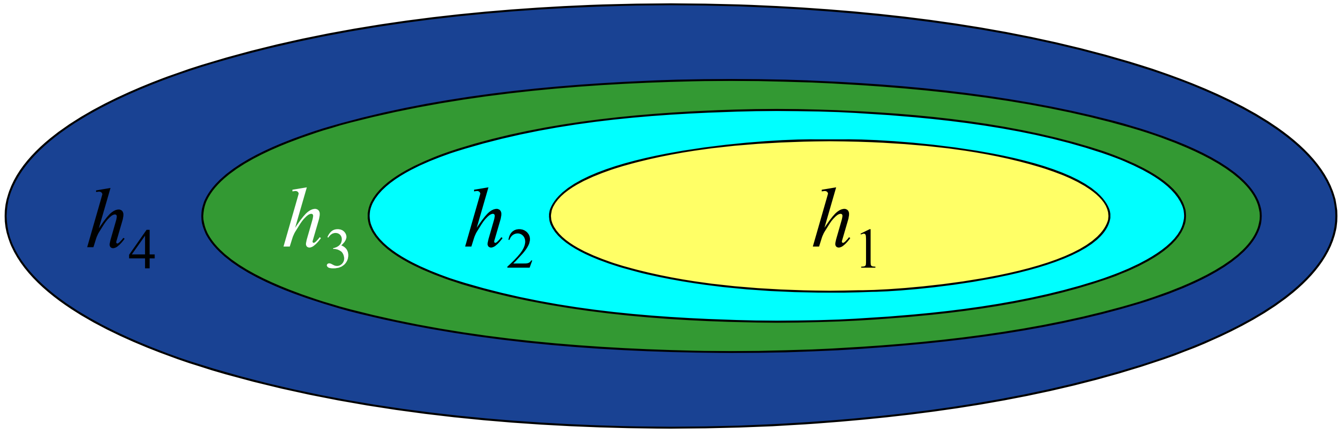
η =0.05 and l=10,000

Amongst machines with zero empirical risk, choose the one with smallest VC dimension

How to evaluate VC dimension?

Structure Risk Minimization

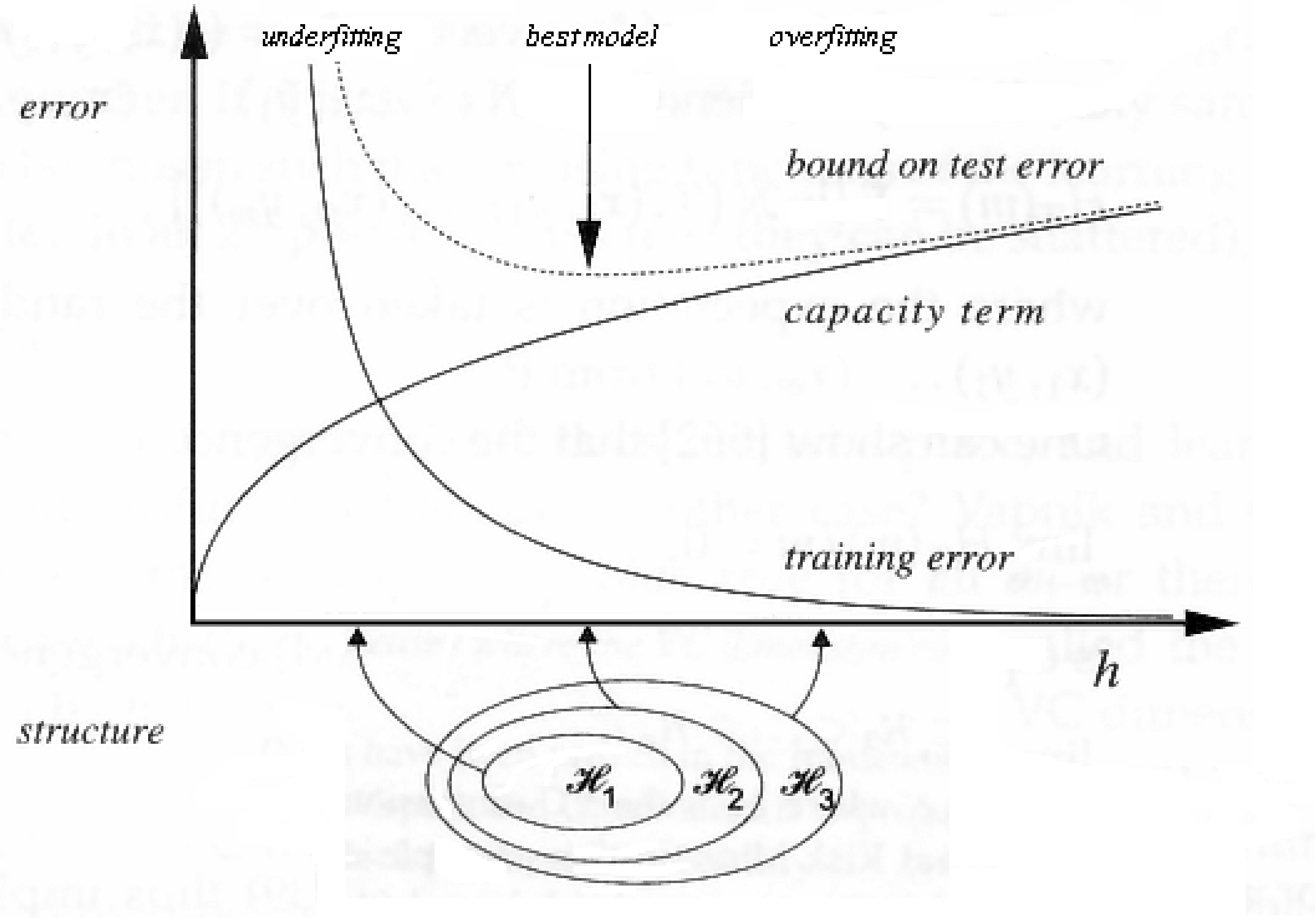
$$h_1 < h_2 < h_3 < h_4$$



Nested subset of functions with different VC dimensions.

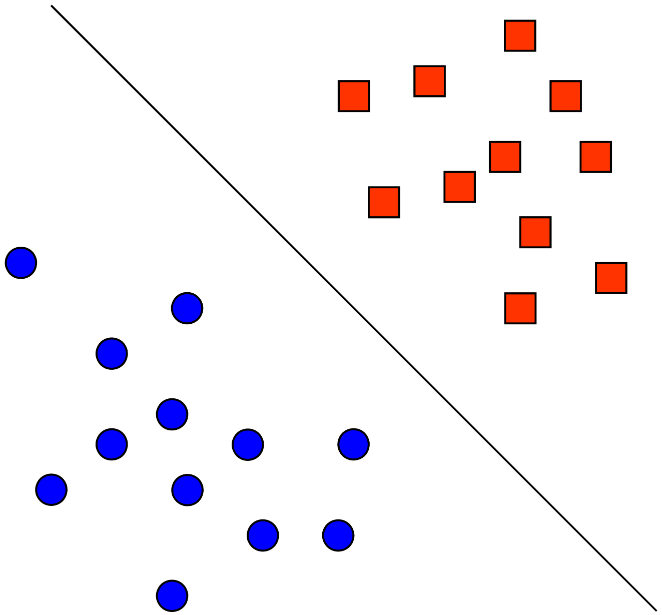
Structure Risk Minimization

MIMA

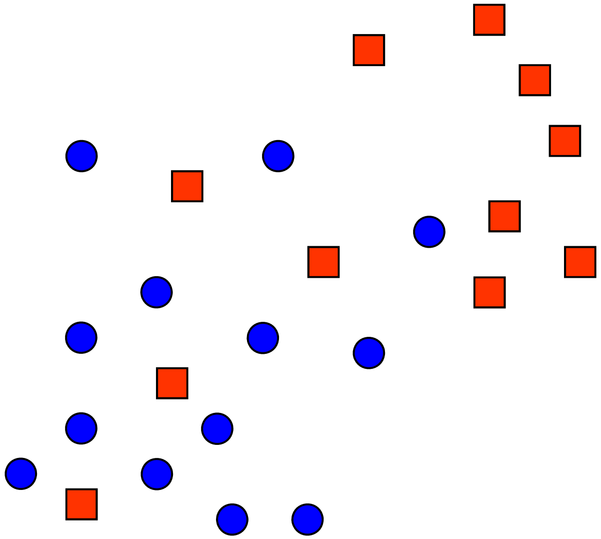


Linear SVM

- The linear separability



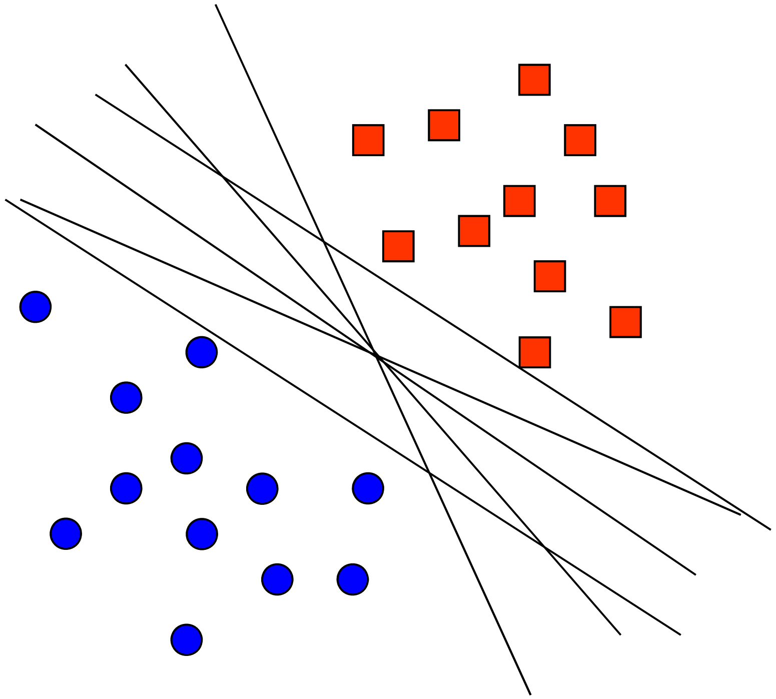
Linearly separable



Not linearly separable

Linear SVM

- The linear separability

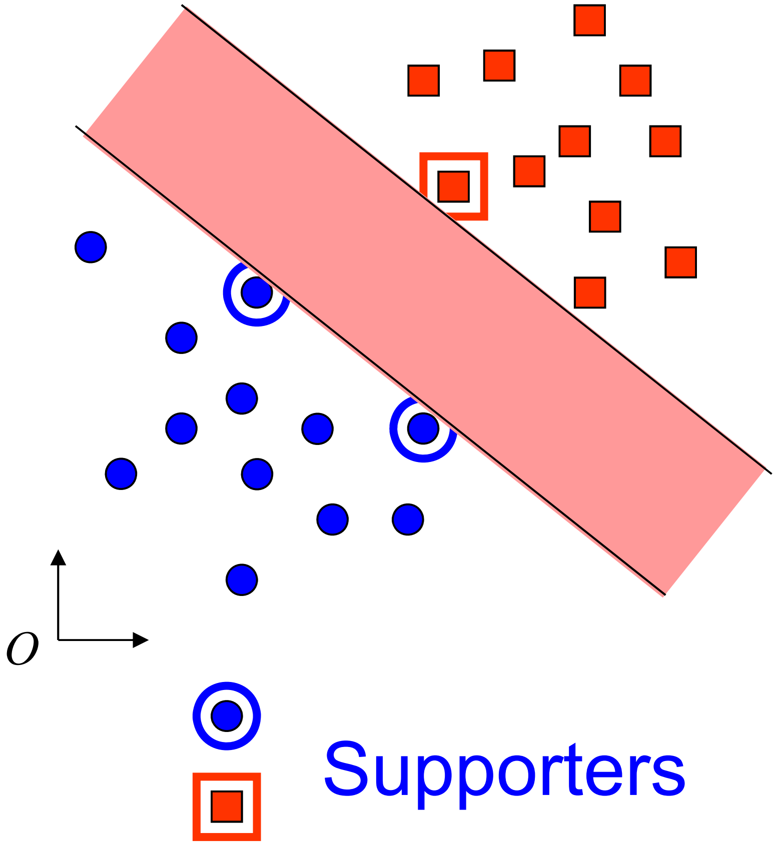


Linearly separable

How would you classify these points using a linear discriminant function in order to minimize the error rate?

Maximum Margin Classifier

$$y_i(\mathbf{wx}_i + b) - 1 \geq 0 \quad \forall i$$



The linear discriminant function (classifier) with the maximum margin is the best

Margin is defined as the width that the boundary could be increased by before hitting a data point

- Why is it the best?
  - Intuitively robust to outliners and thus strong generalization ability

Relation Between VC Dimension and Margin

- What is the relation btw. the margin width and VC dimension?
- Let x belong to sphere of radius R. The set of -margin separating hyperplanes has VC dimension h bounded by:

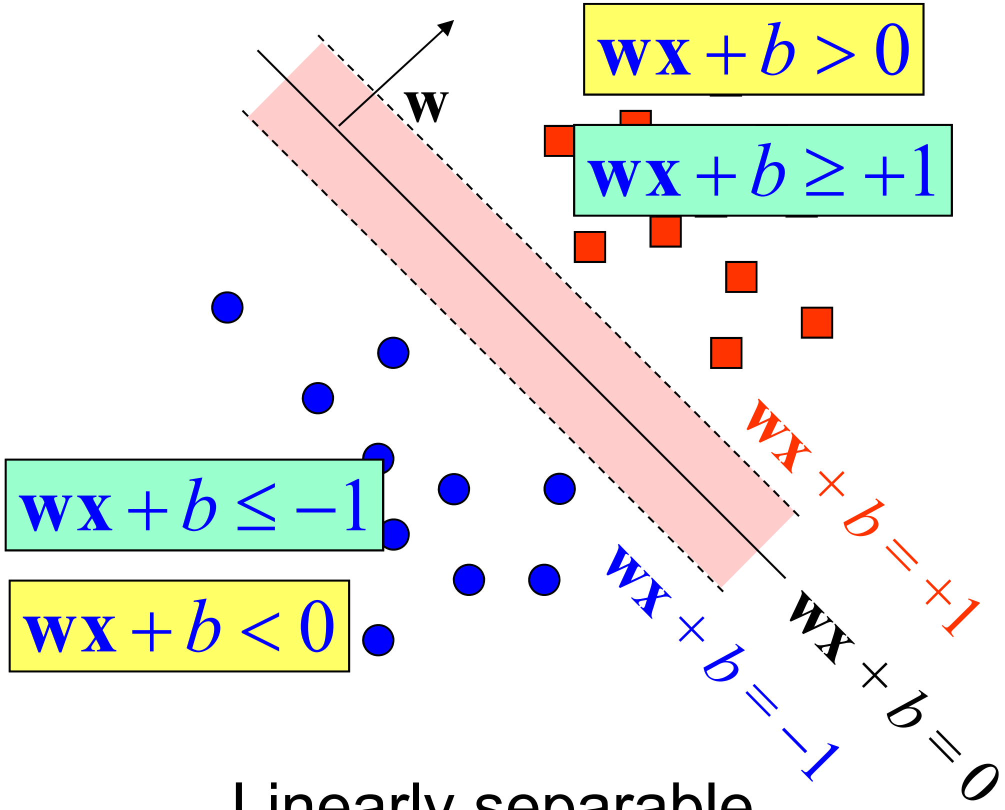
$$h \leq min\left(\left(\frac{R}{\gamma}\right)^2, d\right) + 1$$

d is the dimension of x,

What does this mean?

Linear SVM

- The linear separability



Linearly separable

Linearly Separable

⇒ $\exists \mathbf{w}, b$ such that

$$\mathbf{wx}_i + b \geq +1 \text{ for } y_i = +1$$

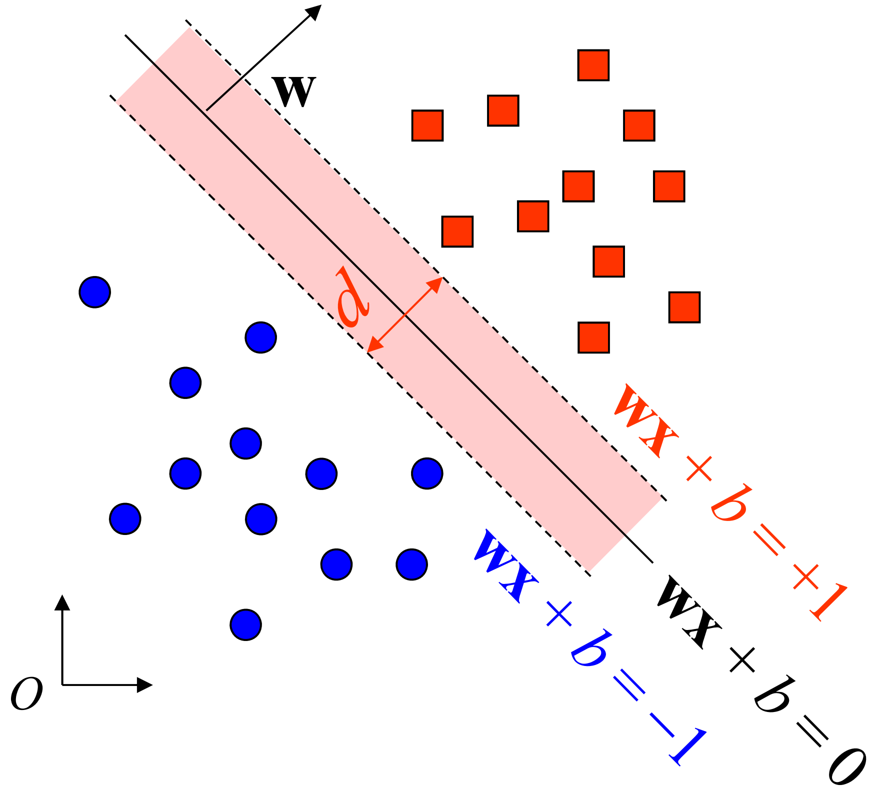
$$\mathbf{wx}_i + b \leq -1 \text{ for } y_i = -1$$

$$\equiv \quad y_i(\mathbf{wx}_i + b) - 1 \geq 0 \quad \forall i$$

Margin Width

MIMA

$$y_i(\mathbf{w}\mathbf{x}_i + b) - 1 \geq 0 \quad \forall i$$



$$d = \frac{1-b}{\|\mathbf{w}\|} - \frac{-1-b}{\|\mathbf{w}\|}$$

$$= \frac{2}{\|\mathbf{w}\|}$$

How about maximize the margin?

Building SVM

Minimize $\frac{1}{2}\|\mathbf{w}\|^2$

Subject to $y_i(\mathbf{w}^T\mathbf{x}_i + b) - 1 \geq 0 \quad \forall i$

This requires the knowledge about Lagrange Multiplier.

The Method of Lagrange

Minimize $\frac{1}{2}\|\mathbf{w}\|^2$

Subject to $y_i(\mathbf{w}^T\mathbf{x}_i + b) - 1 \geq 0 \quad \forall i$

The Lagrangian:

$$L(\mathbf{w}, b; \Lambda) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{l} \lambda_i \left[y_i(\mathbf{w}^T\mathbf{x}_i + b) - 1 \right] \qquad \lambda_i \geq 0$$

Minimize it w.r.t $\mathbf{w}$ **&** $\mathbf{b}$, while *maximize* it w.r.t. Λ.

The Method of Lagrange

- Why Lagrange?
  - The constraints will be replaced by constraints on the Lagrange multipliers, which will be much easier to handle.
  - In this reformulation of the problem, the training data will only appear in the form of dot products between vectors.

The Method of Lagrange

Minimize $\frac{1}{2}\|\mathbf{w}\|$

How about if it is zero?

Subject to $y_i(\mathbf{w}^T\mathbf{x}_i+b)-1\ge 0 \quad \forall i$

What value of λ_i should be if it is feasible and nonzero?

$$L(\mathbf{w},b;\Lambda)=\frac{1}{2}\|\mathbf{w}\|^2-\sum_{i=1}^{l}\lambda_i\left[y_i(\mathbf{w}^T\mathbf{x}_i+b)-1\right] \qquad \lambda_i\ge 0$$

Minimize it w.r.t **w & b**, while *maximize* it w.r.t. Λ.

The Method of Lagrange

$$L(\mathbf{w},b;\Lambda)=\frac{1}{2}\|\mathbf{w}\|^2-\sum_{i=1}^{l}\lambda_i y_i(\mathbf{w}^T\mathbf{x}_i+b)+\sum_{i=1}^{l}\lambda_i$$

Minimize $\frac{1}{2}\|\mathbf{w}\|^2$

Subject to $y_i(\mathbf{w}^T\mathbf{x}_i+b)-1\geq 0 \quad \forall i$

The Lagrangian:

$$L(\mathbf{w},b;\Lambda)=\frac{1}{2}\|\mathbf{w}\|^2-\sum_{i=1}^{l}\lambda_i\left[y_i(\mathbf{w}^T\mathbf{x}_i+b)-1\right] \qquad \lambda_i\geq 0$$

$$=\frac{1}{2}\|\mathbf{w}\|^2-\sum_{i=1}^{l}\lambda_i y_i(\mathbf{w}^T\mathbf{x}_i+b)+\sum_{i=1}^{l}\lambda_i$$

Duality

$$L(\mathbf{w}, b; \Lambda) = \frac{1}{2} \| \mathbf{w} \|^2 - \sum_{i=1}^{l} \lambda_i y_i (\mathbf{w}^T \mathbf{x}_i + b) + \sum_{i=1}^{l} \lambda_i$$

Minimize $\frac{1}{2} \| \mathbf{w} \|^2$

Subject to $y_i (\mathbf{w}^T \mathbf{x}_i + b) - 1 \geq 0 \quad \forall i$

Maximize $L(\mathbf{w}^*, b^*; \Lambda)$

Subject to $\nabla_{\mathbf{w}, b} L(\mathbf{w}, b; \Lambda) = \mathbf{0}$

$\lambda_i \geq 0, \quad i = 1, \mathrm{K}\ , l$

Duality

$$L(\mathbf{w}, b; \Lambda) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{l} \lambda_i y_i (\mathbf{w}^T \mathbf{x}_i + b) + \sum_{i=1}^{l} \lambda_i$$

$$\nabla_{\mathbf{w}} L(\mathbf{w}, b; \Lambda) = \mathbf{w} - \sum_{i=1}^{l} \lambda_i y_i \mathbf{x}_i = \mathbf{0} \quad \Rightarrow \quad \mathbf{w}^* = \sum_{i=1}^{l} \lambda_i y_i \mathbf{x}_i$$

$$\nabla_b L(\mathbf{w}, b; \Lambda) = \sum_{i=1}^{l} \lambda_i y_i = 0 \quad \Rightarrow \quad \sum_{i=1}^{l} \lambda_i y_i = 0$$

Maximize $L(\mathbf{w}^*, b^*; \Lambda)$

Subject to $\nabla_{\mathbf{w}, b} L(\mathbf{w}, b; \Lambda) = \mathbf{0}$

$\lambda_i \geq 0, \quad i = 1, \mathrm{K}\ , l$

Duality

$$L(\mathbf{w},b;\Lambda)=\frac{1}{2}\|\mathbf{w}\|^2-\sum_{i=1}^{l}\lambda_i y_i(\mathbf{w}^T\mathbf{x}_i+b)+\sum_{i=1}^{l}\lambda_i$$

$$\nabla_{\mathbf{w}}L(\mathbf{w},b;\Lambda)=\mathbf{w}-\sum_{i=1}^{l}\lambda_i y_i\mathbf{x}_i=\mathbf{0}\quad\Longrightarrow\quad\mathbf{w}^*=\sum_{i=1}^{l}\lambda_i y_i\mathbf{x}_i$$

$$\nabla_{b}L(\mathbf{w},b;\Lambda)=\sum_{i=1}^{l}\lambda_i y_i=0\quad\Longrightarrow\quad\sum_{i=1}^{l}\lambda_i y_i=0$$

$$L(\mathbf{w}^*,b^*;\Lambda)=\frac{1}{2}\left(\sum_{i=1}^{l}\lambda_i y_i\mathbf{x}_i\right)^T\sum_{i=1}^{l}\lambda_i y_i\mathbf{x}_i-\left(\sum_{i=1}^{l}\lambda_i y_i\mathbf{x}_i\right)^T\sum_{i=1}^{l}\lambda_i y_i\mathbf{x}_i-b\sum_{i=1}^{l}\lambda_i y_i+\sum_{i=1}^{l}\lambda_i$$

$$=\sum_{i=1}^{l}\lambda_i-\frac{1}{2}\left(\sum_{i=1}^{l}\lambda_i y_i\mathbf{x}_i\right)^T\sum_{i=1}^{l}\lambda_i y_i\mathbf{x}_i$$

$$=\sum_{i=1}^{l}\lambda_i-\frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\lambda_i\lambda_j y_i y_j<\mathbf{x}_i,\mathbf{x}_j>\quad\Longleftarrow\quad\text{Maximize}$$

Duality

MIMA

$$L(\mathbf{w},b;\Lambda)=\frac{1}{2}\|\mathbf{w}\|^2-\sum_{i=1}^{l}\lambda_i y_i(\mathbf{w}^T\mathbf{x}_i+b)+\sum_{i=1}^{l}\lambda_i$$

$$\nabla_{\mathbf{w}}L(\mathbf{w},b;\Lambda)=\mathbf{w}-\sum_{i=1}^{l}\lambda_i y_i\mathbf{x}_i=\mathbf{0}\quad\Rightarrow\quad\mathbf{w}^*=\sum_{i=1}^{l}\lambda_i y_i\mathbf{x}_i$$

$$\nabla_{b}L(\mathbf{w},b;\Lambda)=\sum_{i=1}^{l}\lambda_i y_i=0\quad\Rightarrow\quad\sum_{i=1}^{l}\lambda_i y_i=0\qquad\boxed{\Lambda^T\mathbf{y}=0}$$

$$L(\mathbf{w}^*,b^*;\Lambda)=\frac{1}{2}\left(\sum_{i=1}^{l}\lambda_i y_i\mathbf{x}_i\right)^T\sum_{i=1}^{l}\lambda_i y_i\mathbf{x}_i-\left(\sum_{i=1}^{l}\lambda_i y_i\mathbf{x}_i\right)^T\sum_{i=1}^{l}\lambda_i y_i\mathbf{x}_i-b\sum_{i=1}^{l}\lambda_i y_i+\sum_{i=1}^{l}\lambda_i$$

$$=\sum_{i=1}^{l}\lambda_i-\frac{1}{2}\left(\sum_{i=1}^{l}\lambda_i y_i\mathbf{x}_i\right)^T\sum_{i=1}^{l}\lambda_i y_i\mathbf{x}_i\qquad\boxed{F(\Lambda)=\Lambda\cdot\mathbf{1}-\frac{1}{2}\Lambda^T D\Lambda}$$

$$=\sum_{i=1}^{l}\lambda_i-\frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\lambda_i\lambda_j y_i y_j<\mathbf{x}_i,\mathbf{x}_j>$$

⇐ Maximize

Duality

The Primal

Minimize $\frac{1}{2}\|\mathbf{w}\|^2$

Subject to $y_i(\mathbf{w}^T\mathbf{x}_i + b) - 1 \geq 0 \quad \forall i$

The Dual

Maximize $F(\Lambda) = \Lambda \cdot \mathbf{1} - \dfrac{1}{2}\Lambda^T D \Lambda$

Subject to $\Lambda^T \mathbf{y} = 0$

$\Lambda \geq \mathbf{0}$

The Solution

Quadratic Programming

Find Λ* by …

$$\Rightarrow \quad \mathbf{w}^* = \sum_{i=1}^{l} \lambda_i^* y_i \mathbf{x}_i \qquad b^* = ?$$

The Dual

Maximize $F(\Lambda) = \Lambda \cdot \mathbf{1} - \frac{1}{2}\Lambda^T D \Lambda$

Subject to $\Lambda^T \mathbf{y} = 0$

$\Lambda \geq \mathbf{0}$

The Solution

Quadratic Programming

Call it a support vector is $\lambda_i > 0$.

Find Λ^* by …

$$\mathbf{w}^* = \sum_{i=1}^{l} \lambda_i^* y_i \mathbf{x}_i$$

$$b^* = y_i - \mathbf{w}^{*T} \mathbf{x}_i, \quad \underbrace{\lambda_i > 0}_{\text{The Karush-Kuhn-Tucker Conditions}}$$

The Lagrangian:

$$L(\mathbf{w}, b; \Lambda) = \frac{1}{2} \| \mathbf{w} \|^2 - \sum_{i=1}^{l} \lambda_i \left[y_i (\mathbf{w}^T \mathbf{x}_i + b) - 1 \right]$$

The Karush-Kuhn-Tucker Conditions

$$L(\mathbf{w}, b; \Lambda) = \frac{1}{2} \| \mathbf{w} \|^2 - \sum_{i=1}^{l} \lambda_i \left[y_i (\mathbf{w}^T \mathbf{x}_i + b) - 1 \right]$$

$$\nabla_{\mathbf{w}} L(\mathbf{w}, b; \Lambda) = \mathbf{w} - \sum_{i=1}^{l} \lambda_i y_i \mathbf{x}_i = \mathbf{0}$$

$$\nabla_b L(\mathbf{w}, b; \Lambda) = \sum_{i=1}^{l} \lambda_i y_i = 0$$

$$y_i (\mathbf{w}^T \mathbf{x}_i + b) - 1 \geq 0, \quad i = 1, K, l$$

$$\lambda_i \geq 0, \quad i = 1, K, l$$

$$\lambda_i \left[y_i (\mathbf{w}^T \mathbf{x}_i + b) - 1 \right] = 0, \quad i = 1, K, l$$

Classification

$$\mathbf{w}^* = \sum_{i=1}^{l} \lambda_i^* y_i \mathbf{x}_i$$

$$f(\mathbf{x}) = \operatorname{sgn}\left(\mathbf{w}^{*T}\ \mathbf{x} + b^*\right)$$

$$= \operatorname{sgn}\left(\sum_{i=1}^{l} \lambda_i^* y_i < \mathbf{x}_i, \mathbf{x} > + b^*\right)$$

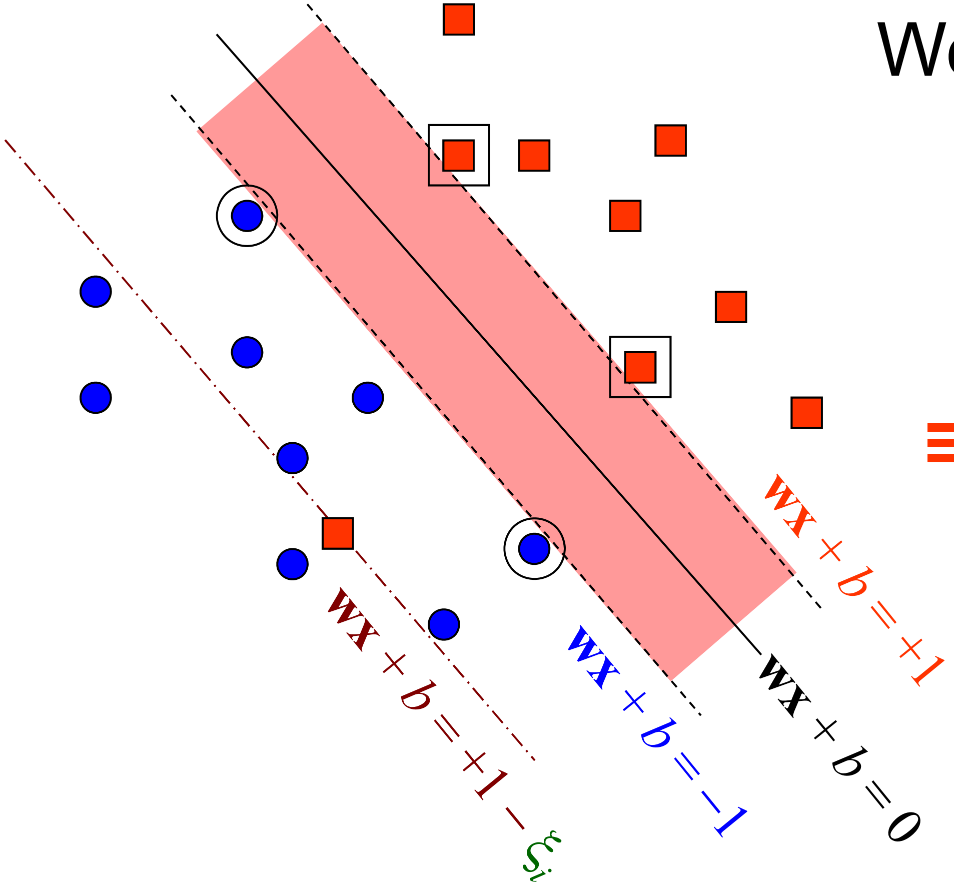
$$= \operatorname{sgn}\left(\sum_{\lambda_i^* \neq 0} \lambda_i^* y_i < \mathbf{x}_i, \mathbf{x} > + b^*\right)$$

Classification Using Supporters

$$f(\mathbf{x}) = \operatorname{sgn}\left(\sum_{\lambda_i^* \neq 0} \overbrace{\lambda_i^* y_i}^{\text{The weight for the } i^{th} \text{ support vector.}} \underbrace{< \mathbf{x}_i , \mathbf{x} >}_{\text{The similarity measure btw. input and the } i^{th} \text{ support vector.}} + \overbrace{b^*}^{\text{Bias}} \right)$$

Linear SVM

- Then non-separable case



We require that

$$\mathbf{wx}_i+b\geq+1-\xi_i \text{ for } y_i=+1$$

$$\mathbf{wx}_i+b\leq-1+\xi_i \text{ for } y_i=-1$$

$$\equiv \quad y_i(\mathbf{wx}_i+b)-1+\xi_i\geq 0 \quad \forall i$$

$$\xi_i\geq 0 \quad \forall i$$

Mathematic Formulation

For simplicity, we consider $k = 1$.

Minimize $\frac{1}{2}\|\mathbf{w}\|^2 + C\left(\sum_i \xi_i\right)^k$

Subject to $y_i(\mathbf{w}^T\mathbf{x}_i + b) - 1 + \xi_i \geq 0 \quad \forall i$

$\xi_i \geq 0 \quad \forall i$

Mathematic Formulation

For simplicity, we consider $k = 1$.

Minimize $\frac{1}{2} \|\mathbf{w}\|^2 + C\left(\sum_i \xi_i\right)^k$

Subject to $y_i(\mathbf{w}^T \mathbf{x}_i + b) - 1 + \xi_i \geq 0 \quad \forall i$

$\xi_i \geq 0 \quad \forall i$

The Lagrangian

Minimize $\frac{1}{2}\|\mathbf{w}\|^2 + C\sum_i \xi_i$

Subject to $y_i(\mathbf{w}^T\mathbf{x}_i + b) - 1 + \xi_i \ge 0 \quad \forall i$

$\xi_i \ge 0 \quad \forall i$

$$L(\mathbf{w}, b, \Xi; \Lambda, \mathrm{M})$$

$$= \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_i \xi_i - \sum_i \lambda_i \left[y_i(\mathbf{w}^T\mathbf{x}_i + b) - 1 + \xi_i \right] - \sum_i \mu_i \xi_i$$

$\lambda_i \ge 0, \mu_i \ge 0$

Duality

$$\boxed{L(\mathbf{w},b,\Xi;\Lambda,\mathrm{M})=\frac{1}{2}\|\mathbf{w}\|^2+C\sum_i\xi_i-\sum_i\lambda_i\left[y_i(\mathbf{w}^T\mathbf{x}_i+b)-1+\xi_i\right]-\sum_i\mu_i\xi_i}$$

$$\boxed{\lambda_i\geq 0,\mu_i\geq 0}$$

Minimize $\frac{1}{2}\|\mathbf{w}\|^2+C\sum_i\xi_i$

Subject to $y_i(\mathbf{w}^T\mathbf{x}_i+b)-1+\xi_i\geq 0 \quad \forall i$

$\xi_i\geq 0 \quad \forall i$

Maximize $L(\mathbf{w}^*,b^*,\Xi^*;\Lambda,\mathrm{M})$

Subject to $\nabla_{\mathbf{w},b,\Xi}L(\mathbf{w},b,\Xi;\Lambda,\mathrm{M})=0$

$\Lambda\geq\mathbf{0},\mathrm{M}\geq\mathbf{0}$

Duality

$\lambda_i \geq 0, \mu_i \geq 0$

$$L(\mathbf{w}, b, \Xi; \Lambda, \mathrm{M}) = \frac{1}{2} \|\mathbf{w}\|^2 + C\sum_i \xi_i - \sum_i \lambda_i \left[y_i(\mathbf{w}^T \mathbf{x}_i + b) - 1 + \xi_i \right] - \sum_i \mu_i \xi_i$$

$$\nabla_{\mathbf{w}} L(\mathbf{w}, b, \Xi; \Lambda, \mathrm{M}) = \mathbf{w} - \sum_i \lambda_i y_i \mathbf{x}_i = \mathbf{0} \quad \Rightarrow \quad \mathbf{w}^* = \sum_i \lambda_i y_i \mathbf{x}_i$$

$$\nabla_b L(\mathbf{w}, b, \Xi; \Lambda, \mathrm{M}) = \sum_i \lambda_i y_i = 0 \quad \Rightarrow \quad \sum_i \lambda_i y_i = 0$$

$$\nabla_{\xi_i} L(\mathbf{w}, b, \Xi; \Lambda, \mathrm{M}) = C - \lambda_i - \mu_i = 0 \quad \Rightarrow \quad \mu_i = C - \lambda_i$$

$$0 \leq \lambda_i \leq C$$

Maximize $L(\mathbf{w}^*, b^*, \Xi^*; \Lambda, \mathrm{M})$

Subject to $\nabla_{\mathbf{w}, b, \Xi} L(\mathbf{w}, b, \Xi; \Lambda, \mathrm{M}) = 0$

$$\Lambda \geq \mathbf{0}, \mathrm{M} \geq \mathbf{0}$$

Duality

$$\lambda_i \geq 0, \mu_i \geq 0$$

$$L(\mathbf{w}, b, \Xi; \Lambda, \mathrm{M}) = \frac{1}{2} \| \mathbf{w} \|^2 + C \sum_i \xi_i - \sum_i \lambda_i \left[y_i (\mathbf{w}^T \mathbf{x}_i + b) - 1 + \xi_i \right] - \sum_i \mu_i \xi_i$$

$$\nabla_{\mathbf{w}} L(\mathbf{w}, b, \Xi; \Lambda, \mathrm{M}) = \mathbf{w} - \sum_i \lambda_i y_i \mathbf{x}_i = \mathbf{0} \Rightarrow \mathbf{w}^* = \sum_i \lambda_i y_i \mathbf{x}_i$$

$$\nabla_b L(\mathbf{w}, b, \Xi; \Lambda, \mathrm{M}) = \sum_i \lambda_i y_i = 0 \Rightarrow \sum_i \lambda_i y_i = 0$$

$$\nabla_{\xi_i} L(\mathbf{w}, b, \Xi; \Lambda, \mathrm{M}) = C - \lambda_i - \mu_i = 0 \Rightarrow \mu_i = C - \lambda_i$$

$$0 \leq \lambda_i \leq C$$

$$F(\Lambda, \mathrm{M}) = L(\mathbf{w}^*, b^*, \Xi^*; \Lambda, \mathrm{M}) = \sum_i \lambda_i - \frac{1}{2} \sum_i \sum_j \lambda_i \lambda_j y_i y_j < \mathbf{x}_i, \mathbf{x}_j >$$

Maximize this

Duality

$$\boxed{\lambda_i \ge 0, \mu_i \ge 0}$$

$$\boxed{L(\mathbf{w},b,\Xi;\Lambda,\mathrm{M}) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_i \xi_i - \sum_i \lambda_i\left[y_i(\mathbf{w}^T\mathbf{x}_i + b) - 1 + \xi_i\right] - \sum_i \mu_i \xi_i}$$

$$\nabla_{\mathbf{w}} L(\mathbf{w},b,\Xi;\Lambda,\mathrm{M}) = \mathbf{w} - \sum_i \lambda_i y_i \mathbf{x}_i = \mathbf{0} \Rightarrow \mathbf{w}^* = \sum_i \lambda_i y_i \mathbf{x}_i$$

$$\nabla_b L(\mathbf{w},b,\Xi;\Lambda,\mathrm{M}) = \sum_i \lambda_i y_i = 0 \Rightarrow \sum_i \lambda_i y_i = 0 \qquad \boxed{\Lambda^T \mathbf{y} = 0}$$

$$\nabla_{\xi_i} L(\mathbf{w},b,\Xi;\Lambda,\mathrm{M}) = C - \lambda_i - \mu_i = 0 \Rightarrow \mu_i = C - \lambda_i$$

$$\boxed{0 \le \Lambda \le C} \quad 0 \le \lambda_i \le C$$

$$F(\Lambda,\mathrm{M}) = L(\mathbf{w}^*,b^*,\Xi^*;\Lambda,\mathrm{M}) = \sum_i \lambda_i - \frac{1}{2}\sum_i \sum_j \lambda_i \lambda_j y_i y_j < \mathbf{x}_i, \mathbf{x}_j >$$

Maximize this

$$\boxed{F(\Lambda) = \Lambda \cdot \mathbf{1} - \frac{1}{2}\Lambda^T D \Lambda}$$

Duality

MIMA

The Primal

Minimize $\frac{1}{2}\|\mathbf{w}\|^2 + C\left(\sum_i \xi_i\right)^k$

Subject to $y_i(\mathbf{w}^T\mathbf{x}_i + b) - 1 + \xi_i \ge 0 \quad \forall i$

$\xi_i \ge 0 \quad \forall i$

The Dual

Maximize $F(\Lambda) = \Lambda \cdot \mathbf{1} - \frac{1}{2}\Lambda^T D \Lambda$

Subject to $\Lambda^T \mathbf{y} = 0$

$\mathbf{0} \le \Lambda \le C\mathbf{1}$

The Karush-Kuhn-Tucker Conditions

$$L(\mathbf{w},b,\Xi;\Lambda,\mathrm{M})=\frac{1}{2}\|\mathbf{w}\|^2+C\sum_i\xi_i-\sum_i\lambda_i\left[y_i(\mathbf{w}^T\mathbf{x}_i+b)-1+\xi_i\right]-\sum_i\mu_i\xi_i$$

$$\nabla_{\mathbf{w}}L(\mathbf{w},b,\Xi;\Lambda,\mathrm{M})=\mathbf{w}-\sum_i\lambda_i y_i\mathbf{x}_i=\mathbf{0}$$

$$\nabla_b L(\mathbf{w},b,\Xi;\Lambda,\mathrm{M})=\sum_i\lambda_i y_i=0$$

$$\nabla_{\xi_i}L(\mathbf{w},b,\Xi;\Lambda,\mathrm{M})=C-\lambda_i-\mu_i=0$$

$$y_i(\mathbf{w}^T\mathbf{x}_i+b)-1+\xi_i\ge 0$$

$$\xi_i\ge 0$$

$$\mu_i\ge 0$$

$$\lambda_i\ge 0$$

$$\lambda_i\left[y_i(\mathbf{w}^T\mathbf{x}_i+b)-1+\xi_i\right]=0$$

$$\mu_i\xi_i=0$$

The Solution

MIMA

Quadratic Programming

Find Λ^* by …

$$\Rightarrow \quad \mathbf{w}^* = \sum_{i=1}^{l} \lambda_i^* y_i \mathbf{x}_i$$

$$b^* = ?$$

$$\Xi = ?$$

The Dual

Maximize $\quad F(\Lambda) = \Lambda \cdot \mathbf{1} - \frac{1}{2}\Lambda^T D \Lambda$

Subject to $\quad \Lambda^T \mathbf{y} = 0$

$$\mathbf{0} \le \Lambda \le C\mathbf{1}$$

The Solution

Find Λ^* by …

Call it a support vector is $0 < \lambda_i < \mathrm{C}$.

$$\mathbf{w}^* = \sum_{i=1}^{l} \lambda_i^* y_i \mathbf{x}_i$$

$$b^* = y_i - \mathbf{w}^{*T} \mathbf{x}_i, \quad 0 < \lambda_i < C$$

The Lagrangian:

$$L(\mathbf{w}, b, \Xi; \Lambda, \mathrm{M})$$

$$= \frac{1}{2} \| \mathbf{w} \|^2 + C \sum_i \xi_i - \sum_i \lambda_i \left[y_i (\mathbf{w}^T \mathbf{x}_i + b) - 1 + \xi_i \right] - \sum_i \mu_i \xi_i$$

The Solution

Find Λ^* by …

Call it a support vector is $0 < \lambda_i < \mathrm{C}$.

$$\mathbf{w}^* = \sum_{i=1}^{l} \lambda_i^* y_i \mathbf{x}_i$$

$$b^* = y_i - \mathbf{w}^{*T}\,\mathbf{x}_i, \quad 0<\lambda_i < C$$

$$\xi_i = \max\left[0, 1 - y_i(\mathbf{w}^*\mathbf{x}_i + b^*)\right]$$

A false classification pattern if $\xi_i > 1$.

Classification

$$\mathbf{w}^* = \sum_{i=1}^{l} \lambda_i^* y_i \mathbf{x}_i$$

$$f(\mathbf{x}) = \operatorname{sgn}\left(\mathbf{w}^{*T}\mathbf{x} + b^*\right)$$

$$= \operatorname{sgn}\left(\sum_{i=1}^{l} \lambda_i^* y_i < \mathbf{x}_i, \mathbf{x} > + b^*\right)$$

$$= \operatorname{sgn}\left(\sum_{\lambda_i^* \neq 0} \lambda_i^* y_i < \mathbf{x}_i, \mathbf{x} > + b^*\right)$$

Classification Using Supporters

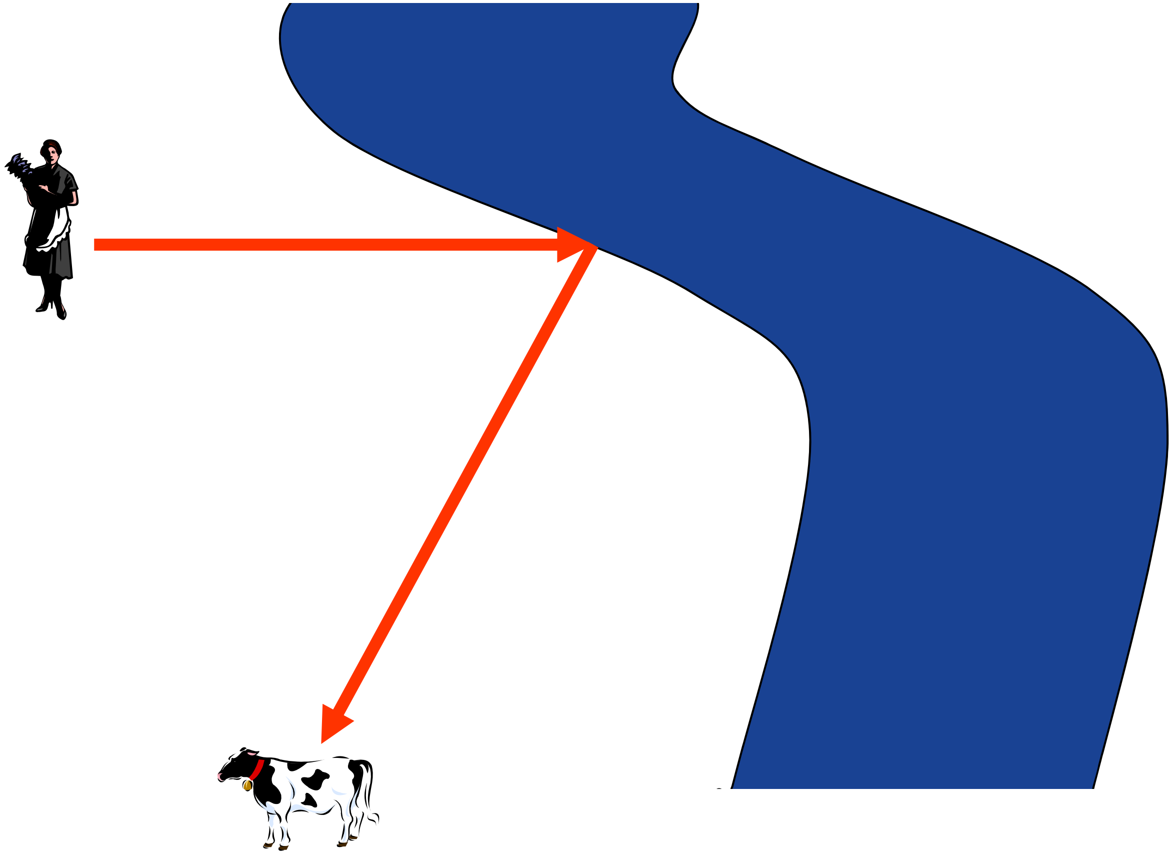
$$f(\mathbf{x}) = \operatorname{sgn}\left(\sum_{\lambda_i^* \neq 0} \overbrace{\lambda_i^* y_i}^{\text{The weight for the } i^{th} \text{ support vector.}} \underbrace{< \mathbf{x}_i, \mathbf{x} >}_{\text{The similarity measure btw. input and the } i^{th} \text{ support vector.}} + \overbrace{b^*}^{\text{Bias}} \right)$$

Lagrange Multiplier

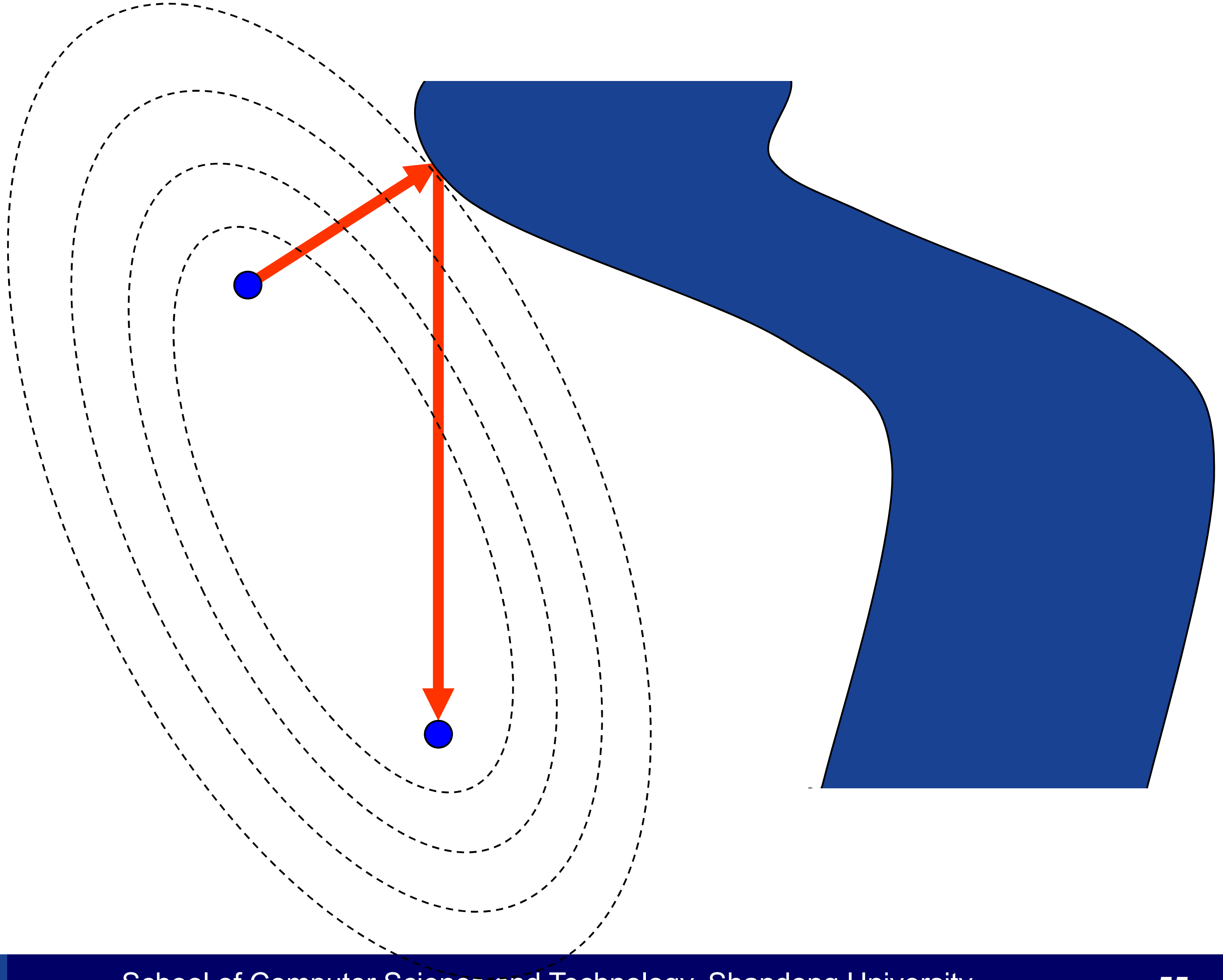
- "Lagrange Multiplier Method" is a powerful tool for constraint optimization.
- Contributed by Riemann.

Milkmaid Problem

MIMA



Milkmaid Problem



Milkmaid Problem

MIMA

$$f(x,y)=\sum_{i=1}^{2}\sqrt{(x-x_i)^2+(y-y_i)^2}$$

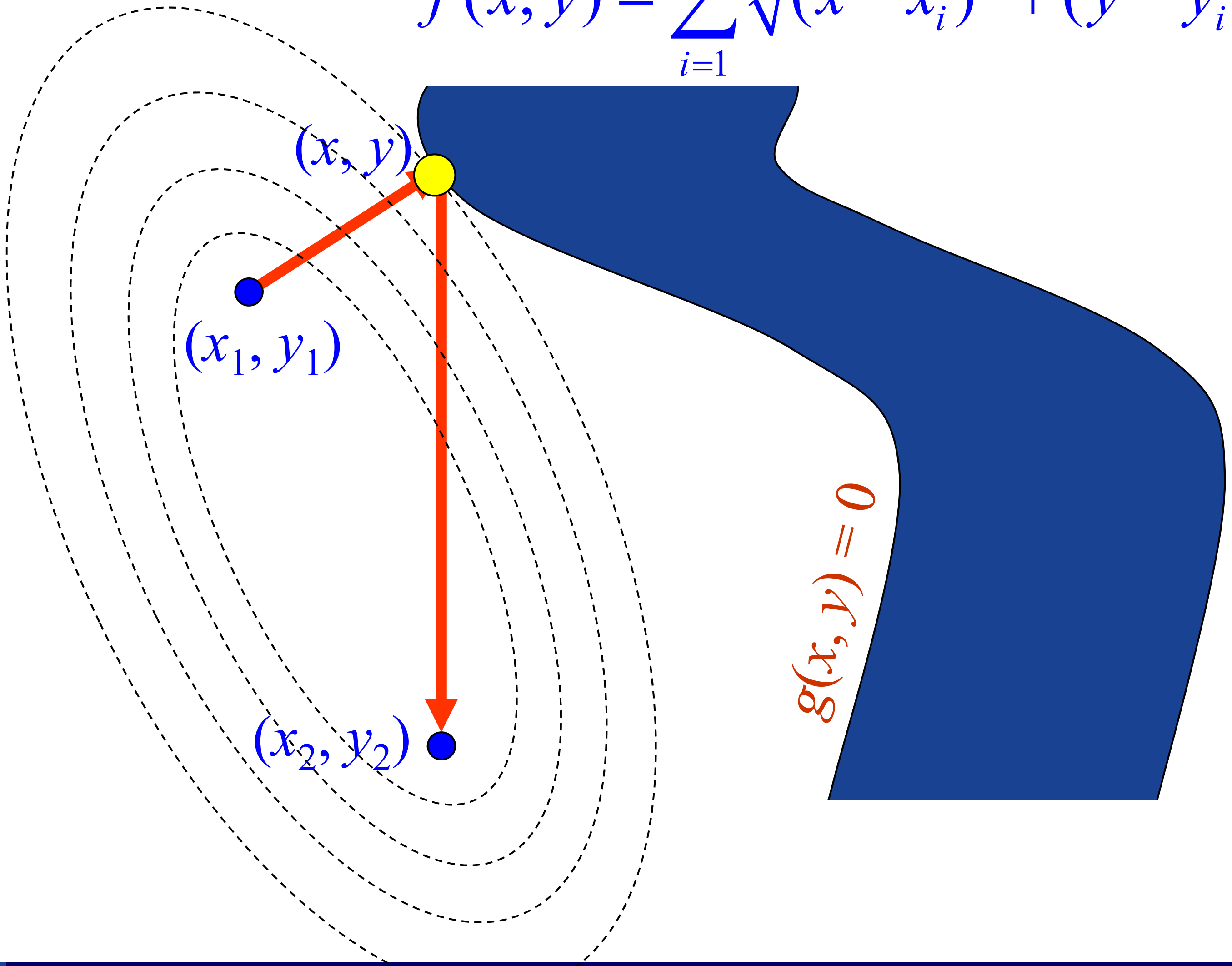
Goal:

Minimize

$f(x,y)$

Subject to

$g(x,y)=0$



Observation

$$f(x,y)=\sum_{i=1}^{2}\sqrt{(x-x_i)^2+(y-y_i)^2}$$

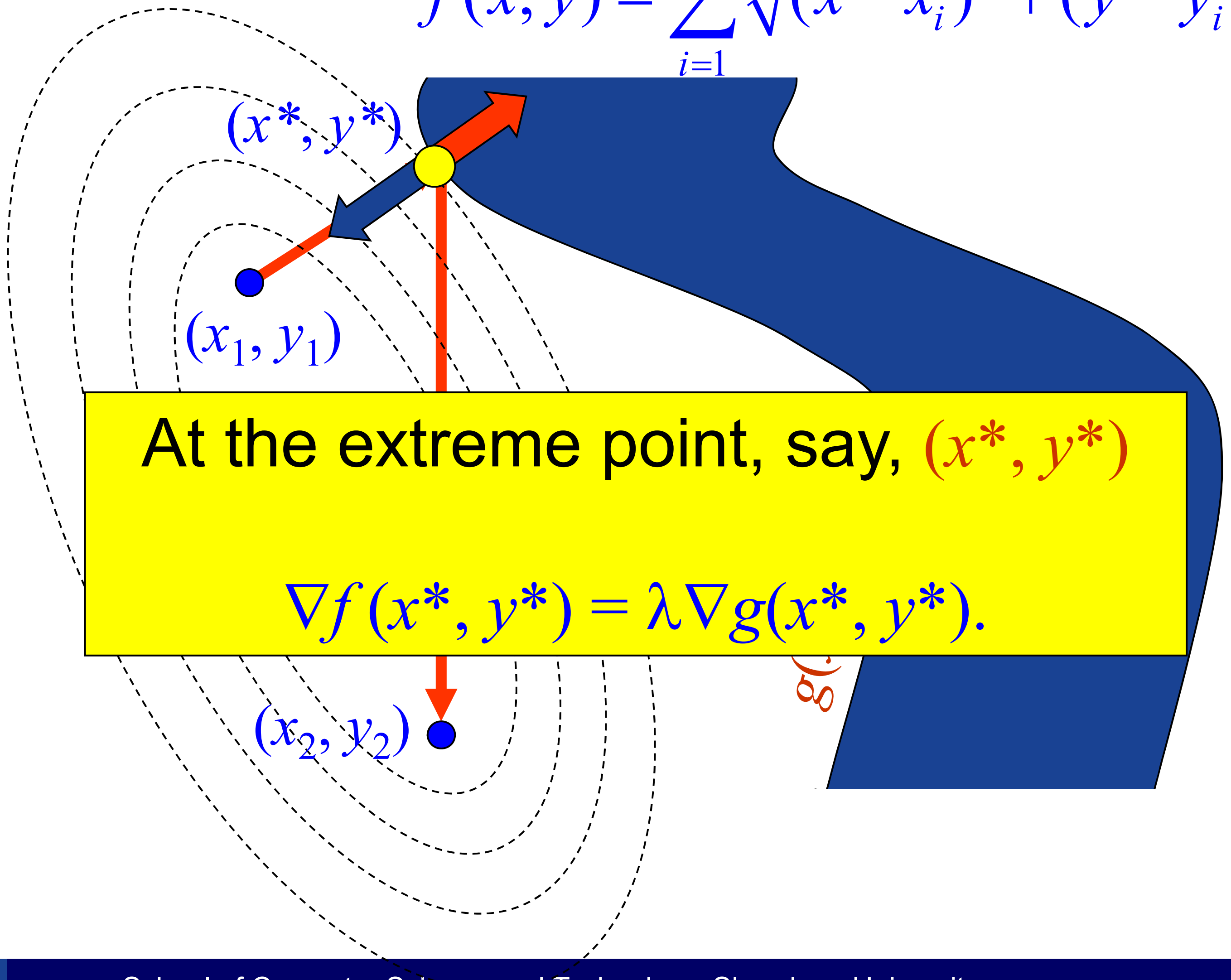
Goal:

Minimize

$$f(x,y)$$

Subject to

$$g(x,y)=0$$



At the extreme point, say, (x^*,y^*)

$$\nabla f(x^*,y^*)=\lambda\nabla g(x^*,y^*).$$

Optimization with Equality Constraints

Goal: Min/Max $f(\mathbf{x})$

Subject to $g(\mathbf{x}) = 0$

Lemma:

At an extreme point, say, $\mathbf{x}^*$, we have

$$\nabla f(\mathbf{x}^*) = \lambda \nabla g(\mathbf{x}^*) \text{ if } \nabla g(\mathbf{x}^*) \neq 0$$

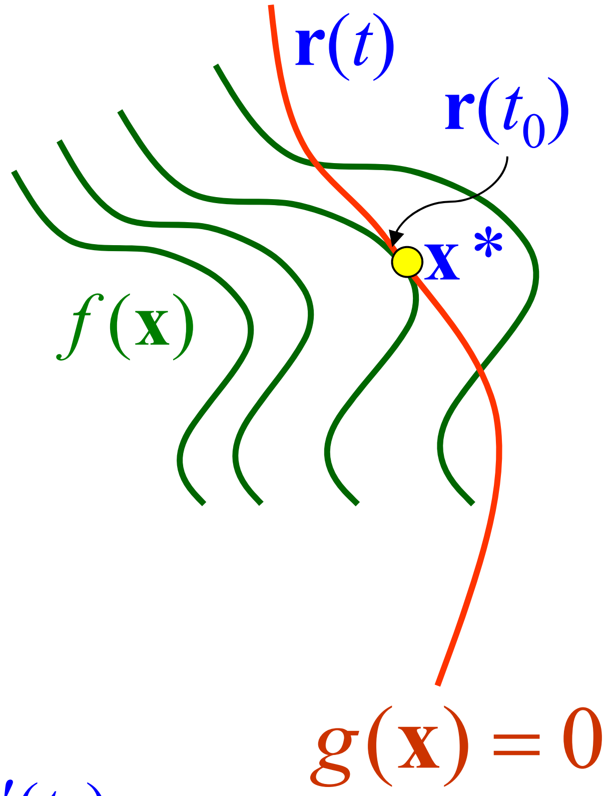
Proof

MIMA

$$\nabla f(\mathbf{x}^*) = \lambda \nabla g(\mathbf{x}^*) \text{ if } \nabla g(\mathbf{x}^*) \neq 0$$

Let $\mathbf{x}^*$ be an extreme point.

$\mathbf{r}(t)$ be any differentiable path on surface $g(\mathbf{x})=0$ such that $\mathbf{r}(t_0)=\mathbf{x}^*$.



➡ $\mathbf{r}'(t_0)$ is a vector tangent to the surface $g(\mathbf{x})=0$ at $\mathbf{x}^*$.

➡ $f(\mathbf{x}^*) = f(\mathbf{r}(t))\,|_{t=t_0}$

$$\tfrac{d}{dt} f(\mathbf{r}(t)) = \nabla f(\mathbf{r}(t)) \cdot \mathbf{r}'(t)$$

Since $\mathbf{x}^*$ be an extreme point,

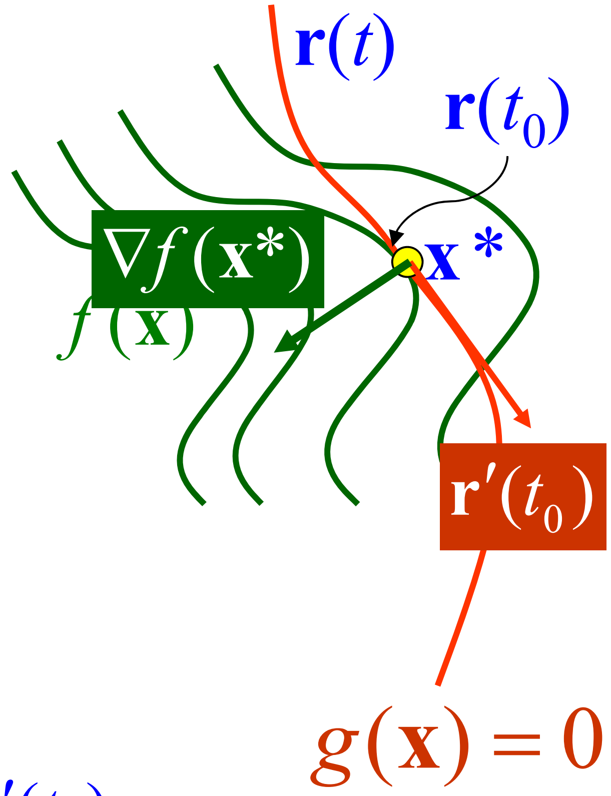
➡ $0 = \nabla f(\mathbf{r}(t_0)) \cdot \mathbf{r}'(t_0) = \nabla f(\mathbf{x}^*) \cdot \mathbf{r}'(t_0)$

Proof

$$\nabla f(\mathbf{x}^*) = \lambda \nabla g(\mathbf{x}^*) \text{ if } \nabla g(\mathbf{x}^*) \neq 0$$

$$\nabla f(\mathbf{x}^*) \perp \mathbf{r}'(t_0)$$

This is true for any $\mathbf{r}$ pass through $\mathbf{x}^*$ on surface $g(\mathbf{x})=0$.



It implies that $$\nabla f(\mathbf{x}^*) \perp \Gamma,$$

where Γ is the *tangential plane* of surface $g(\mathbf{x})=0$ at $\mathbf{x}^*$.

$$0 = \nabla f(\mathbf{r}(t_0)) \cdot \mathbf{r}'(t_0) = \nabla f(\mathbf{x}^*) \cdot \mathbf{r}'(t_0)$$

Optimization with Equality Constraints

Goal: Min/Max $f(\mathbf{x})$

Subject to $g(\mathbf{x}) = 0$

Lemma:

At an extreme point, say, $\mathbf{x}^*$, we have

$$\nabla f(\mathbf{x}^*) = \lambda \nabla g(\mathbf{x}^*) \text{ if } \nabla g(\mathbf{x}^*) \neq 0$$

Lagrange Multiplier

The Method of Lagrange

$\mathbf{x}$: dimension n.

Goal: Min/Max $f(\mathbf{x})$

Subject to $g(\mathbf{x}) = 0$

Find the extreme points by solving the following equations.

$$\left.\begin{aligned} \nabla f(\mathbf{x}) &= \lambda \nabla g(\mathbf{x}) \\ g(\mathbf{x}) &= 0 \end{aligned}\right\}$$

$n+1$ equations
with $n+1$ variables

Lagrangian

MIMA

Goal:

$$\left.\begin{array}{ll}\text{Min/Max} & f(\mathbf{x}) \\ \text{Subject to} & g(\mathbf{x}) = 0\end{array}\right\} \textit{Constraint}\ \text{Optimization}$$

Define $L(\mathbf{x};\lambda) = f(\mathbf{x}) + \lambda g(\mathbf{x})$ —— Lagrangian

Solve

$$\left.\begin{array}{l}\nabla_{\mathbf{x}} L(\mathbf{x};\lambda) = \mathbf{0} \\ \nabla_{\lambda} L(\mathbf{x};\lambda) = \mathbf{0}\end{array}\right\} \textit{Unconstraint}\ \text{Optimization}$$

Optimization with Multiple Equality Constraints

$$\Lambda = (\lambda_1, \ldots, \lambda_m)^T$$

Min/Max $\quad f(\mathbf{x})$

Subject to $\quad g_i(\mathbf{x}) = 0, \quad i = 1, \ldots, m$

Define $\quad L(\mathbf{x}; \Lambda) = f(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i g_i(\mathbf{x})$ — Lagrangian

Solve $\quad \nabla_{\mathbf{x}, \Lambda} L(\mathbf{x}; \Lambda) = \mathbf{0}$

Optimization with Inequality Constraints

Minimize $f(\mathbf{x})$

Subject to $g_i(\mathbf{x}) = 0, \quad i = 1, \ldots, m$

$$h_j(\mathbf{x}) \le 0, \quad j = 1, \ldots, n$$

You can always reformulate your problems into the about form.

Lagrange Multipliers

$$\Lambda = (\lambda_1, \ldots, \lambda_m)^T$$

$$\mathrm{M} = (\mu_1, \ldots, \mu_n)^T \qquad \mu_i \geq 0$$

Minimize $\quad f(\mathbf{x})$

Subject to $\quad g_i(\mathbf{x}) = 0, \quad i = 1, \ldots, m$

$$h_j(\mathbf{x}) \leq 0, \quad j = 1, \ldots, n$$

Lagrangian:

$$L(\mathbf{x}; \Lambda, \mathrm{M}) = f(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i g_i(\mathbf{x}) + \sum_{j=1}^{n} \mu_j h_j(\mathbf{x})$$

Lagrange Multipliers

$$\Lambda = (\lambda_1, K, \lambda_m)^T$$

$$M = (\mu_1, K, \mu_n)^T \qquad \mu_i \geq 0$$

Minimize $f(\mathbf{x})$

Subject to $g_i(\mathbf{x}) = 0, \quad i = 1, K, m$

$h_j(\mathbf{x}) \leq 0, \quad j = 1, K, n$

negative for feasible solutions

0 for feasible solutions

Lagrangian:

$$L(\mathbf{x}; \Lambda, M) = f(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i g_i(\mathbf{x}) + \sum_{j=1}^{n} \mu_j h_j(\mathbf{x})$$

Duality

$$L(\mathbf{x};\Lambda,\mathrm{M}) = f(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i g_i(\mathbf{x}) + \sum_{j=1}^{n} \mu_j h_j(\mathbf{x})$$

Let $\mathbf{x}^*$ be a local extreme.

$$\Rightarrow \quad \mathbf{0} = \nabla_{\mathbf{x}} L(\mathbf{x};\Lambda,\mathrm{M})\,|_{\mathbf{x}=\mathbf{x}^*}$$

Define $D(\Lambda,\mathrm{M}) = L(\mathbf{x}^*;\Lambda,\mathrm{M})$

$$D(\Lambda,\mathrm{M}) = f(\mathbf{x}^*) + \sum_{i=1}^{m} \lambda_i g_i(\mathbf{x}^*) + \sum_{j=1}^{n} \mu_j h_j(\mathbf{x}^*) \le f(\mathbf{x}^*)$$

Maximize it w.r.t. Λ, M

Duality

$$L(\mathbf{x};\Lambda,\mathrm{M}) = f(\ \ )\ \sum^{m}\ \ \sum^{n}$$

To *minimize* the Lagrangian w.r.t $\mathbf{x}$, while to *maximize* it w.r.t. Λ and M.

Let $\mathbf{x}^*$ be a local extreme.

$$\mathbf{0} = \nabla_{\mathbf{x}} L(\mathbf{x};\Lambda,\mathrm{M})\,|_{\mathbf{x}=\mathbf{x}^*}$$

What are we doing?

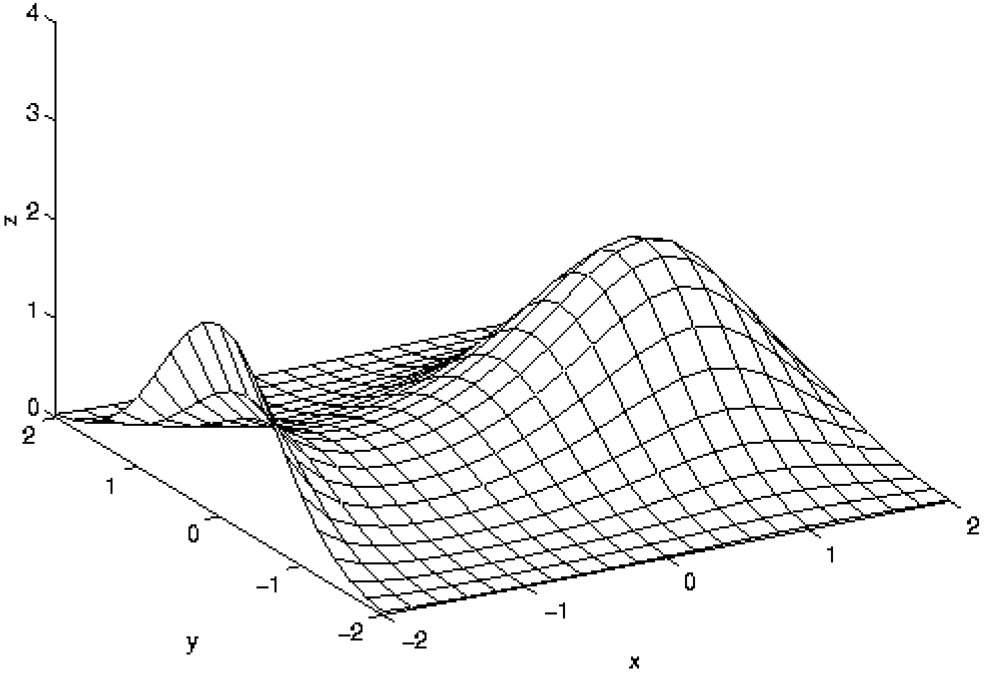
Define $D(\Lambda,\mathrm{M}) = L(\mathbf{x}^*;\Lambda,\mathrm{M})$

$$D(\Lambda,\mathrm{M}) = f(\mathbf{x}^*) + \sum_{i=1}^{m} \lambda_i g_i(\mathbf{x}^*) + \sum_{j=1}^{n} \mu_j h_j(\mathbf{x}^*) \ \le f(\mathbf{x}^*)$$

Maximize it w.r.t. Λ, M

Saddle Point Determination

$$L(\mathbf{x};\Lambda,\mathrm{M}) = f(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i g_i(\mathbf{x}) + \sum_{j=1}^{n} \mu_j h_j(\mathbf{x})$$



Saddle Point Determination

$$L(\mathbf{x};\Lambda,\mathrm{M}) = f(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i g_i(\mathbf{x}) + \sum_{j=1}^{n} \mu_j h_j(\mathbf{x})$$

The primal

Minimize $f(\mathbf{x})$

Subject to $g_i(\mathbf{x}) = 0, \quad i = 1, \mathrm{K}\ , m$

$h_j(\mathbf{x}) \le 0, \quad j = 1, \mathrm{K}\ , n$

The dual

Maximize $L(\mathbf{x}^*;\Lambda,\mathrm{M})$

Subject to $\nabla_{\mathbf{x},\Lambda} L(\mathbf{x};\Lambda,\mathrm{M}) = \mathbf{0}$

$\mathrm{M} \ge \mathbf{0}$

The Karush-Kuhn-Tucker Conditions

$$L(\mathbf{x};\Lambda,\mathrm{M}) = f(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i g_i(\mathbf{x}) + \sum_{j=1}^{n} \mu_j h_j(\mathbf{x})$$

$$\nabla_{\mathbf{x}} L(\mathbf{x};\Lambda,\mathrm{M}) = \nabla_{\mathbf{x}} f(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i \nabla_{\mathbf{x}} g_i(\mathbf{x}) + \sum_{j=1}^{n} \mu_j \nabla_{\mathbf{x}} h_j(\mathbf{x}) = \mathbf{0}$$

$$g_i(\mathbf{x}) = 0, \quad i = 1,\mathrm{K}\ ,m$$

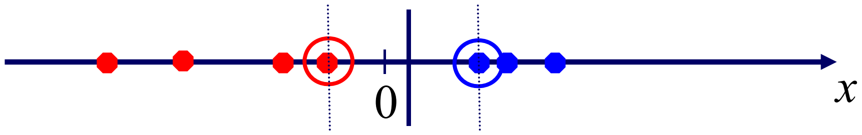
$$\mu_j \geq 0, \quad j = 1,\mathrm{K}\ ,n$$

$$h_j(\mathbf{x}) \leq 0, \quad j = 1,\mathrm{K}\ ,n$$

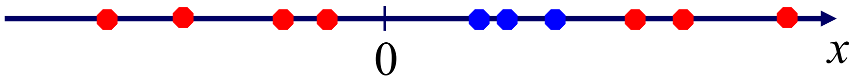
$$\mu_j h_j(\mathbf{x}) = 0, \quad j = 1,\mathrm{K}\ ,n$$

Non-linear SVMs

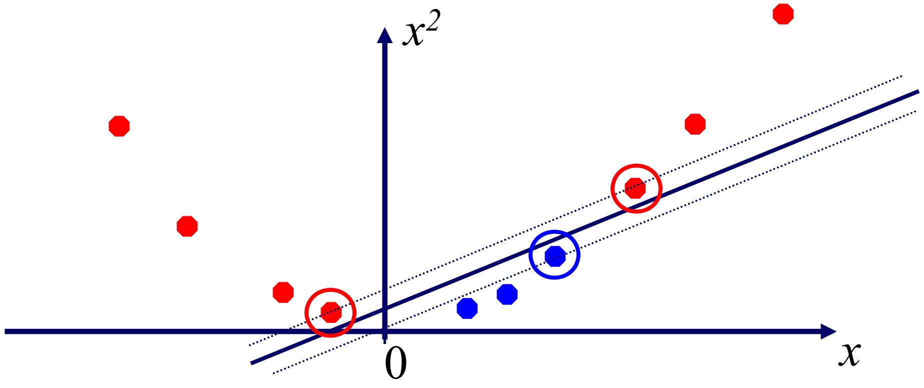
- □ Datasets that are linearly separable with noise work out great:



- □ But what are we going to do if the dataset is just too hard?

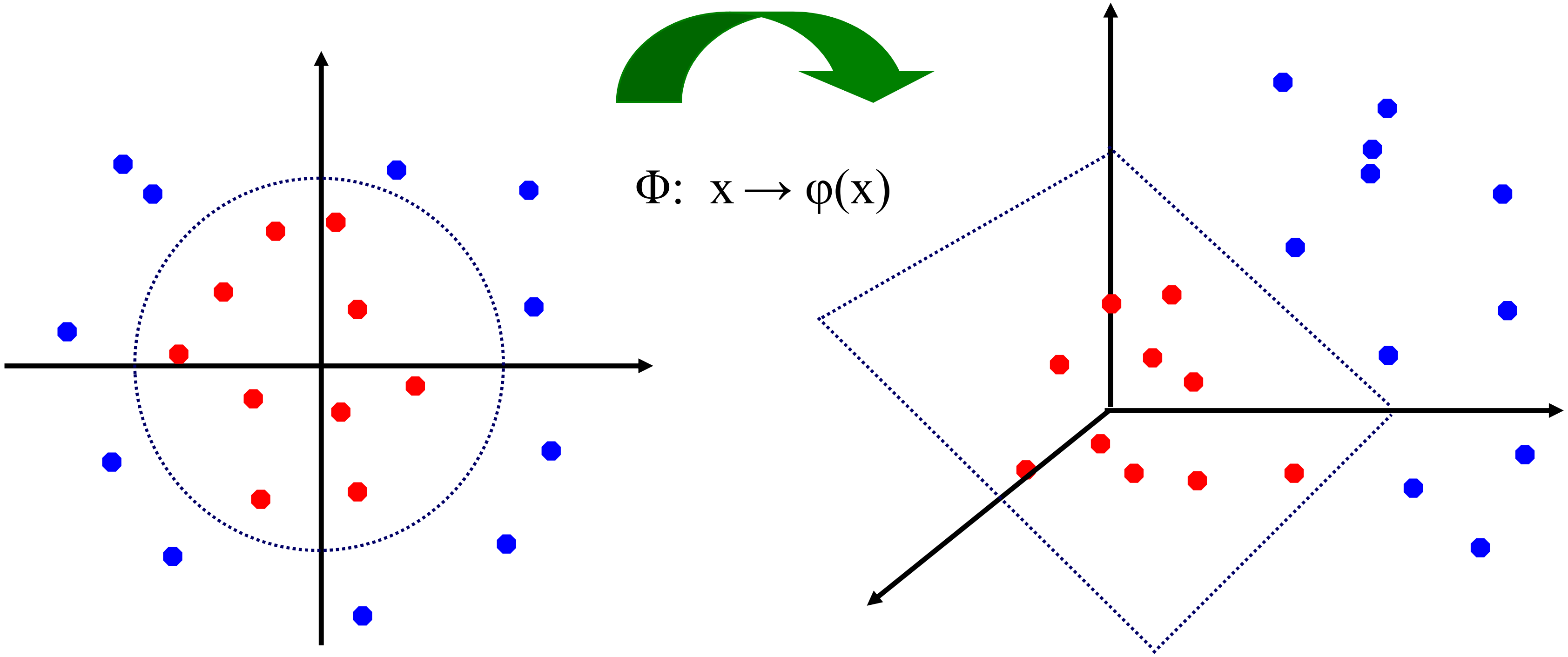


- □ How about… mapping data to a higher-dimensional space:



Non-linear SVMs: Feature Space

- □ General idea: the original input space can be mapped to some higher-dimensional feature space where the training set is separable:



$\Phi: \ x \rightarrow \varphi(x)$

Nonlinear SVMs: The Kernel Trick

- With this mapping, our discriminant function is now:

$$g(x) = w^T \phi(x) + b = \sum_{x_i \in SV} \lambda_i y_i \boxed{\phi(x_i)\phi(x)} + b$$

- No need to know this mapping explicitly, because we only use the dot product of feature vectors in both the training and test.

- A *kernel function* is defined as a function that corresponds to a dot product of two feature vectors in some expanded feature space:

$$K(\mathbf{x}_i, \mathbf{x}_j) \equiv \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$$

Nonlinear SVMs: The Kernel Trick

□ An example:

2-dimensional vectors $\mathrm{x}=[x_1 \;\; x_2]$;

let $K(\mathrm{x_i},\mathrm{x_j})=(1+\mathrm{x_i}^{\mathrm{T}}\mathrm{x_j})^2$,

Need to show that $K(\mathrm{x_i},\mathrm{x_j}) = \varphi(\mathrm{x_i})^{\mathrm{T}}\varphi(\mathrm{x_j})$:

$$K(\mathrm{x_i},\mathrm{x_j})=(1+\mathrm{x_i}^{\mathrm{T}}\mathrm{x_j})^2,$$

$$= 1+ x_{i1}^2x_{j1}^2 + 2\, x_{i1}x_{j1}\, x_{i2}x_{j2}+ x_{i2}^2x_{j2}^2 + 2x_{i1}x_{j1} + 2x_{i2}x_{j2}$$

$$= [1 \;\; x_{i1}^2 \;\; \sqrt{2}\, x_{i1}x_{i2} \;\; x_{i2}^2 \;\; \sqrt{2}x_{i1} \;\; \sqrt{2}x_{i2}]\, [1 \;\; x_{j1}^2 \;\; \sqrt{2}\, x_{j1}x_{j2} \;\; x_{j2}^2 \;\; \sqrt{2}x_{j1} \;\; \sqrt{2}x_{j2}]^{\mathrm{T}}$$

$= \varphi(\mathrm{x_i})^{\mathrm{T}}\varphi(\mathrm{x_j})$, where $\varphi(\mathrm{x}) = [1 \;\; x_1^2 \;\; \sqrt{2}\, x_1x_2 \;\; x_2^2 \;\; \sqrt{2}x_1 \;\; \sqrt{2}x_2]^{\mathrm{T}}$

Nonlinear SVMs: The Kernel Trick

- Examples of commonly-used kernel functions:
  - Linear kernel: $K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \mathbf{x}_j$
  - Polynomial kernel: $K(\mathbf{x}_i, \mathbf{x}_j) = (1 + \mathbf{x}_i^T \mathbf{x}_j)^p$
  - Gaussian (Radial-Basis Function (RBF)) kernel:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\frac{\left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2}{2\sigma^2})$$

  - Sigmoid:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \tanh(\beta_0 \mathbf{x}_i^T \mathbf{x}_j + \beta_1)$$

- In general, functions that satisfy *Mercer's condition* can be kernel functions.

Nonlinear SVM: Optimization

□ Formulation: (Lagrangian Dual Problem)

$$\max \sum_i \lambda_i - \frac{1}{2}\sum_i \sum_j \lambda_i \lambda_j y_i y_j K(x_i, x_j)$$

such that $0 \le \lambda_i \le C$

$$\sum_i \lambda_i y_i = 0$$

□ The solution of the discriminant function is

$$g(x) = w^T \phi(x) + b = \sum_{x_i \in SV} \lambda_i y_i K(x, x_i) + b$$

□ The optimization technique is the same.

Support Vector Machine: Algorithm

- 1. Choose a kernel function
- 2. Choose a value for *C*
- 3. Solve the quadratic programming problem (many software packages available)
- 4. Construct the discriminant function from the support vectors

Other issues

- Choice of kernel

  - Gaussian or polynomial kernel is default

  - if ineffective, more elaborate kernels are needed

  - domain experts can give assistance in formulating appropriate similarity measures

- Choice of kernel parameters

  - e.g. σ in Gaussian kernel

  - σ is the distance between closest points with different classifications

  - In the absence of reliable criteria, applications rely on the use of a validation set or cross-validation to set such parameters.

- Optimization criterion – Hard margin v.s. Soft margin

  - a lengthy series of experiments in which various parameters are tested

Comparison with Neural Networks

- Neural Networks
  - Hidden Layers map to lower dimensional spaces
  - Search space has multiple local minima
  - Training is expensive
  - Classification extremely efficient
  - Requires number of hidden units and layers
  - Very good accuracy in typical domains

- SVMs
  - Kernel maps to a very-high dimensional space
  - Search space has a unique minimum
  - Training is extremely efficient
  - Classification extremely efficient
  - Kernel and cost the two parameters to select
  - Very good accuracy in typical domains
  - Extremely robust

- UCI datasets: http://archive.ics.uci.edu/ml/datasets.html
  - Reuters-21578 Text Categorization Collection
  - Wine
  - Credit Approval
- Requirements
  - Use different kernels(>=3)
  - Choose best values for parameters
  - You can also use dimension reduction method, e.g., PCA

- 针对UCI数据集（http://archive.ics.uci.edu/ml/datasets.html）中的Musk(version2), Wine，采用三种SVM来对其进行分类，计算准确率。其中每种SVM要求用不同的核函数。另外，采用一种集成学习方法，将不同模型集成，集成的模型可以是不同核函数的SVM，也可以加上神经网络、KNN、线性模型、多项式模型等。比较集成模型与SVM模型及其他模型的结果。
- 要求：6月17日24时之前提交代码和报告。

【 Thank You ! 】

Any Question?