Pattern Information Processing: Support Vector Machines

Masashi Sugiyama
(Department of Computer Science)

Contact: W8E-505
sugi@cs.titech.ac.jp
http://sugiyama-www.cs.titech.ac.jp/~sugi/

(Binary) Classification Problem

- Output values are $y_i = \pm 1$
- We want to predict whether output values of unlearned input points are positive/negative.

- Multi-class problem can be transferred to several binary classification problems:
  - One-versus-rest
  - One-versus-one

(Binary) Classification Problem

- In classification, we may still use the same learning methods, e.g., quadratically-constrained least-squares:

$$\hat{\boldsymbol{\alpha}}_{QCLS} = \underset{\boldsymbol{\alpha}\in\mathbb{R}^b}{\operatorname{argmin}} \left[J_{LS}(\boldsymbol{\alpha}) + \lambda \langle \boldsymbol{R}\boldsymbol{\alpha}, \boldsymbol{\alpha} \rangle \right]$$

$$\lambda \ (\geq 0)$$

$$J_{LS}(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \left(\hat{f}(\boldsymbol{x}_i) - y_i \right)^2$$

- Prediction:

$$\widehat{y} = \operatorname{sign}\left(\hat{f}(\boldsymbol{x}) \right)$$

0/1-Loss

- In classification, only the sign of the learned function is used.
- It is natural to use 0/1-loss instead of squared-loss $J_{LS}(\boldsymbol{\alpha})$:

$$J_{0/1}(\boldsymbol{\alpha}) = \sum_{i=1}^{n} I\left(\mathrm{sign}(\hat{f}(\boldsymbol{x}_i)) \neq y_i\right)$$

$$I(a \neq b) = \begin{cases} 0 & (a = b) \\ 1 & (a \neq b) \end{cases}$$

- $J_{0/1}(\boldsymbol{\alpha})$ corresponds to the number of misclassified samples (thus natural).

Hinge-Loss

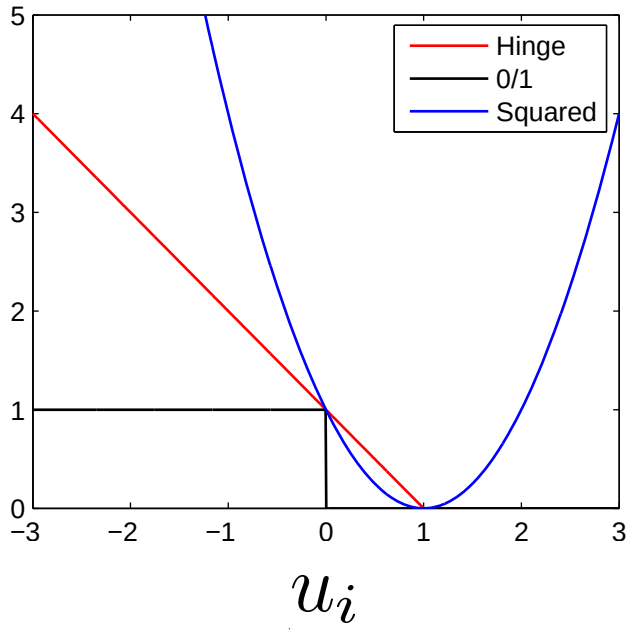
- However, $J_{0/1}(\boldsymbol{\alpha})$ is non-convex so we may not be able to obtain the global minimizer.
- Use hinge-loss as an approximation:

$$J_H(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \max(0, 1 - u_i)$$

$$J_{0/1}(\boldsymbol{\alpha}) = \frac{1}{2} \sum_{i=1}^{n} (1 - \text{sign}(u_i))$$

$$J_{LS}(\boldsymbol{\alpha}) = \sum_{i=1}^{n} (1 - u_i)^2$$

$$u_i = \hat{f}(\boldsymbol{x}_i) y_i$$



Note : $y_i^2 = 1,\ 1/y_i = y_i$

How to Obtain Solutions

$$\hat{\boldsymbol{\alpha}}_{SVM} = \underset{\boldsymbol{\alpha}\in\mathbb{R}^b}{\operatorname{argmin}} \left[J_H(\boldsymbol{\alpha}) + \lambda \langle \boldsymbol{R}\boldsymbol{\alpha}, \boldsymbol{\alpha} \rangle \right]$$

$$J_H(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \max\left(0, 1 - u_i\right)$$

- How to deal with "max"? Use following lemma:

Lemma:

$$\max(0, 1-u) = \min_{\xi\in\mathbb{R}} \xi \quad \text{subject to } \xi \geq 1-u$$
$$\xi \geq 0$$

- Proof: Constraints are $\xi \geq \max(0, 1-u)$,

so the lemma holds. Q.E.D.

How to Obtain Solutions (cont.)

- So we have

$$J_H(\boldsymbol{\alpha}) = \min_{\boldsymbol{\xi} \in \mathbb{R}^n} \langle \mathbf{1}_n, \boldsymbol{\xi} \rangle \ \text{ subject to } \boldsymbol{\xi} \geq \mathbf{1}_n - \boldsymbol{u}$$
$$\boldsymbol{\xi} \geq \mathbf{0}_n$$

- Then $\hat{\boldsymbol{\alpha}}_{SVM}$ is given as

$$\hat{\boldsymbol{\alpha}}_{SVM} = \underset{\boldsymbol{\alpha} \in \mathbb{R}^b, \boldsymbol{\xi} \in \mathbb{R}^n}{\operatorname{argmin}} \ [\langle \mathbf{1}_n, \boldsymbol{\xi} \rangle + \lambda \langle \boldsymbol{R}\boldsymbol{\alpha}, \boldsymbol{\alpha} \rangle]$$
$$\text{subject to } \boldsymbol{\xi} \geq \mathbf{1}_n - \boldsymbol{u}$$
$$\boldsymbol{\xi} \geq \mathbf{0}_n$$

Support Vector Machines

- We focus on the following setting:
  - $\widehat{f}(\boldsymbol{x}) = \sum_{i=1}^{n} \alpha_i K(\boldsymbol{x}, \boldsymbol{x}_i)$
  - $\boldsymbol{R} = \boldsymbol{K}$ $\qquad \boldsymbol{K}_{i,j} = K(\boldsymbol{x}_i, \boldsymbol{x}_j)$
- Putting $\lambda = (2C)^{-1}$ (convention), we have

$$\widehat{\boldsymbol{\alpha}}_{SVM} = \underset{\boldsymbol{\alpha}, \boldsymbol{\xi} \in \mathbb{R}^n}{\operatorname{argmin}} \left[C\langle \mathbf{1}_n, \boldsymbol{\xi} \rangle + \frac{1}{2} \langle \boldsymbol{K}\boldsymbol{\alpha}, \boldsymbol{\alpha} \rangle \right]$$
$$\text{subject to } \boldsymbol{\xi} \geq \mathbf{1}_n - \boldsymbol{u}$$
$$\boldsymbol{\xi} \geq \mathbf{0}_n$$

$$u_i = \widehat{f}(\boldsymbol{x}_i) y_i$$

Efficient Formulation

- The SVM solution can be obtained by

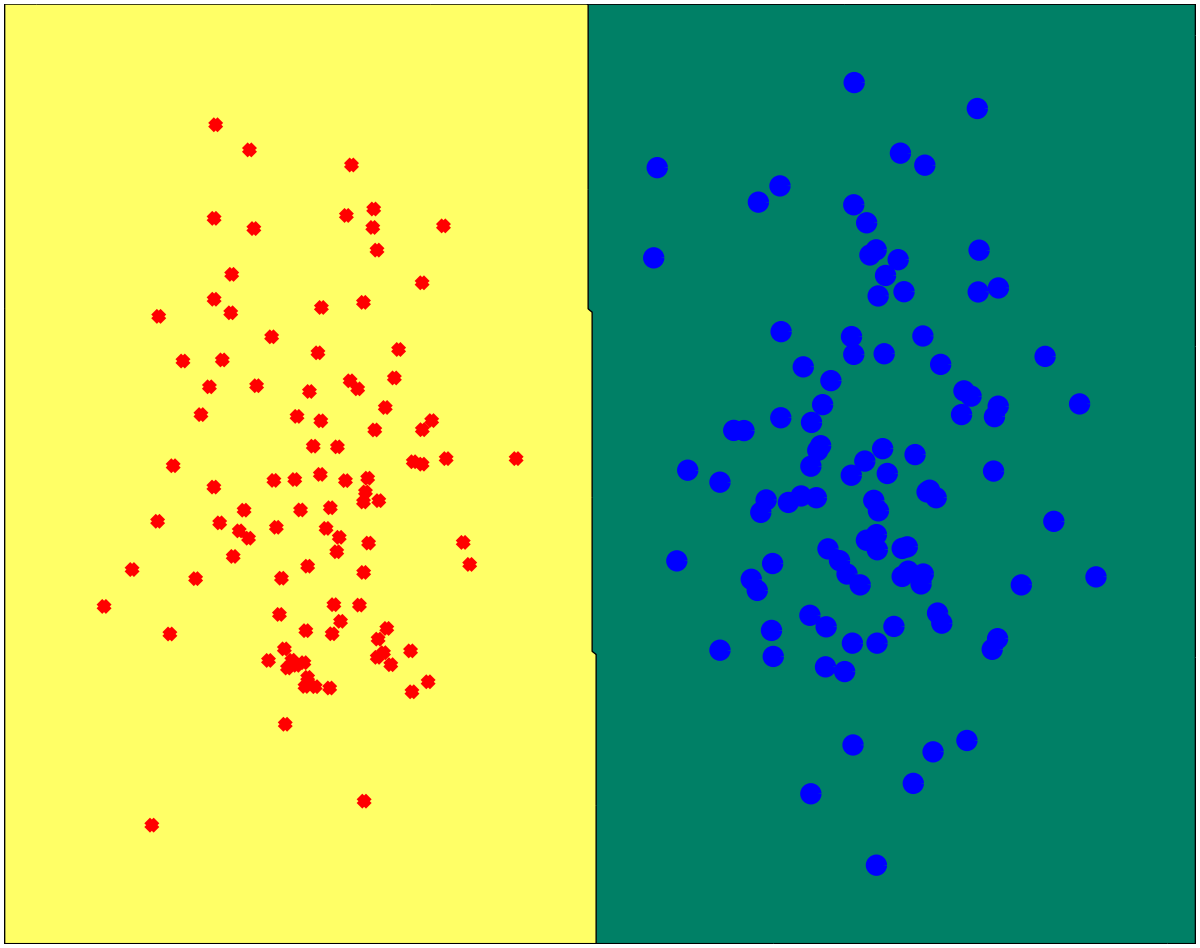
$$[\widehat{\boldsymbol{\alpha}}_{SVM}]_i = [\widehat{\boldsymbol{\beta}}_{SVM}]_i y_i$$

$$\widehat{\boldsymbol{\beta}}_{SVM} = \underset{\boldsymbol{\beta}\in\mathbb{R}^n}{\operatorname{argmax}} \left[\sum_{i=1}^{n} \beta_i - \frac{1}{2} \sum_{i,j=1}^{n} \beta_i \beta_j y_i y_j \boldsymbol{K}_{i,j} \right]$$

$$\text{subject to } \mathbf{0}_n \leq \boldsymbol{\beta} \leq C\mathbf{1}_n$$

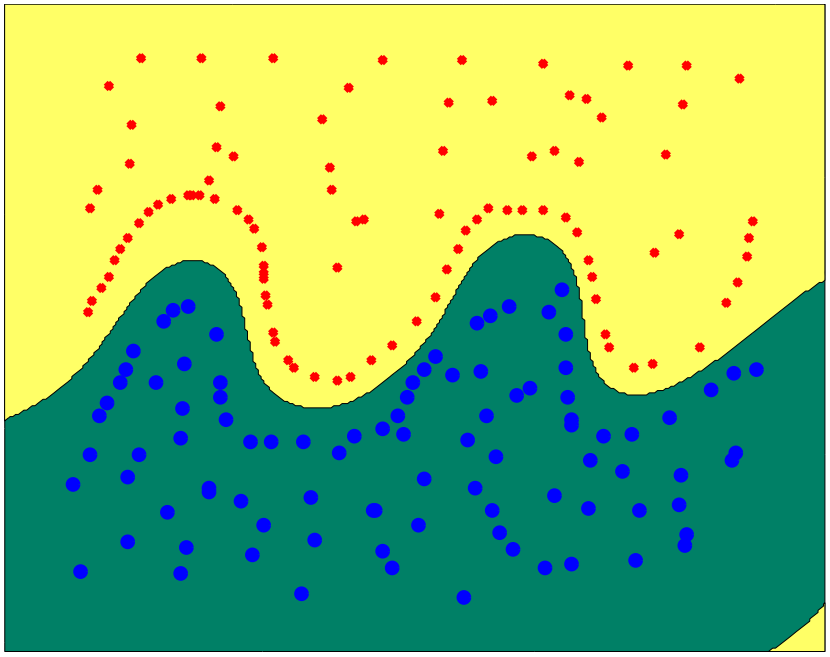
- The number of parameters is decreased from $2n$ to n !
- This corresponds to considering the Wolfe dual (details are omitted).

Examples

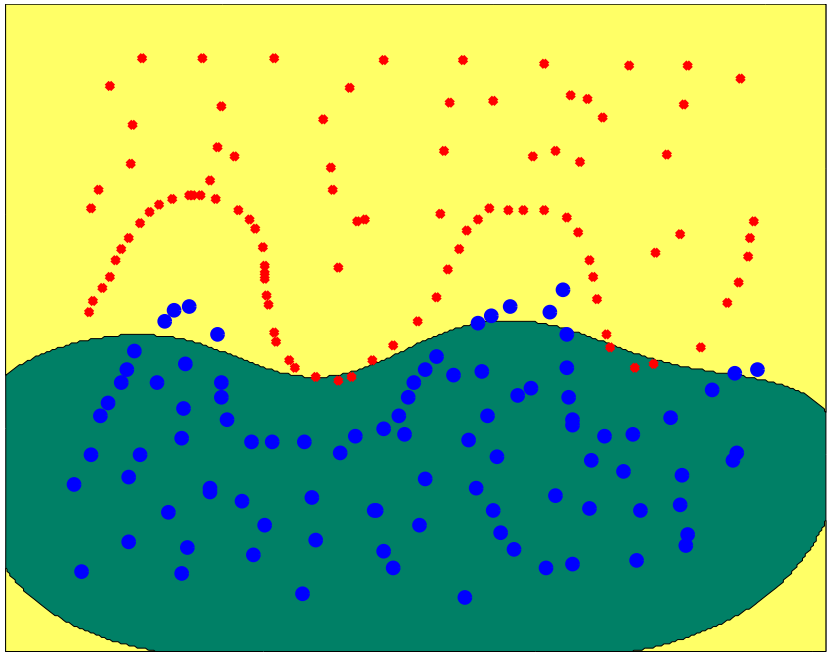


Gaussian kernel: $K(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x}'\|^2}{2c^2}\right)$

Examples (cont.)

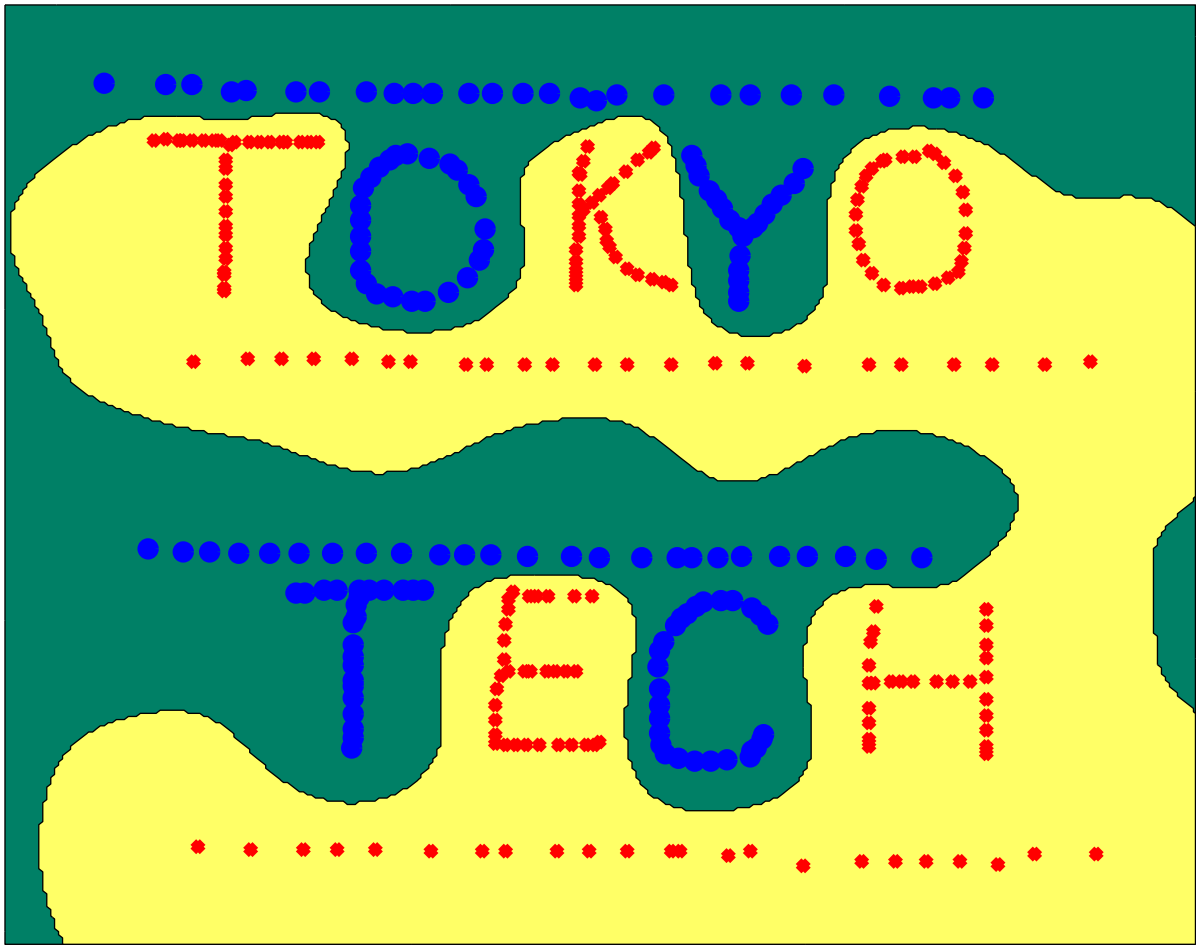


Large C



Small C

Examples



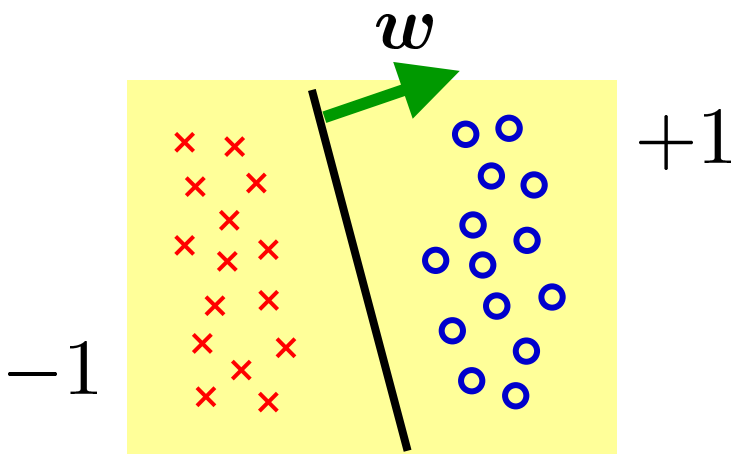
Original Derivation of SVMs

- The way SVMs were introduced today is quite different from the original derivation.
- Let's briefly follow the original derivation.
  - Hyper-plane classifier
  - VC theory
  - Margin maximization
  - Soft margin
  - Kernel trick

Hyper-plane Classifier

- Separate sample space by hyper-plane.

$$\widehat{f}(\boldsymbol{x}) = \operatorname{sign}(\langle \boldsymbol{w}, \boldsymbol{x} \rangle + b)$$

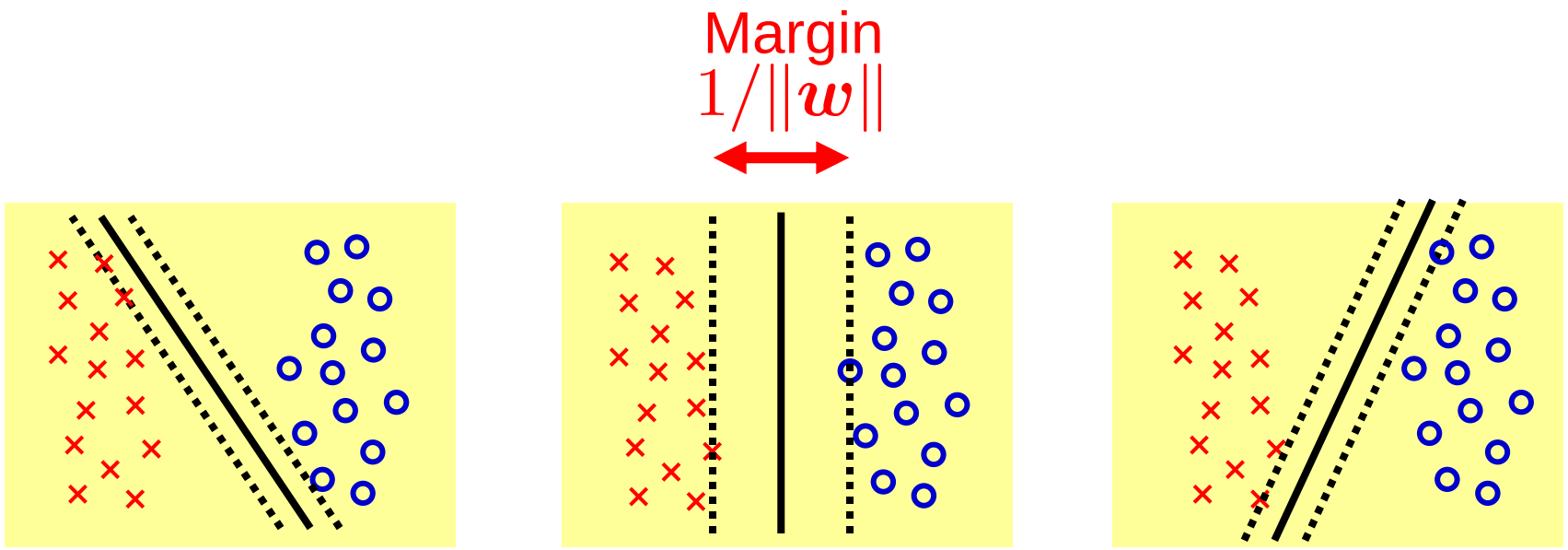


find $\boldsymbol{w}, b$

such that $y_i \widehat{f}(\boldsymbol{x}_i) \geq 1$ for $i = 1, \ldots, n$.

Margin

- Margin: “Gap” between two classes



Small margin | Large margin | Small margin

Vapnik-Chevonenkis Theory

- Generalization error:

$$R[\widehat{f}] = \int\int I(\widehat{f}(\boldsymbol{x}) \neq y) p(\boldsymbol{x}, y) d\boldsymbol{x} dy$$

- Empirical error:

$$R_{\mathrm{emp}}[\widehat{f}] = \frac{1}{n} \sum_{i=1}^{n} I(\widehat{f}(\boldsymbol{x}_i) \neq y_i)$$

$$I(a \neq b) = \begin{cases} 0 & (a = b) \\ 1 & (a \neq b) \end{cases}$$

- Generalization error bound ("VC bound")

$$R[\widehat{f}] \leq R_{\mathrm{emp}}[\widehat{f}] + \sqrt{\frac{1}{n}\left(h\left(\log\frac{2n}{h}+1\right)+\log\frac{4}{\delta}\right)}$$

with probability $1-\delta$

h : VC dimension (model complexity)

Vapnik-Chevonenkis Theory (cont.)

- VC bound:

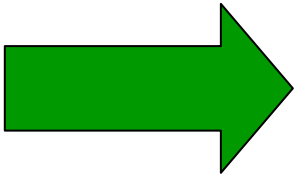
$$R[\widehat{f}] \leq R_{\mathrm{emp}}[\widehat{f}] + \sqrt{\frac{1}{n}\left(h\left(\log\frac{2n}{h}+1\right)+\log\frac{4}{\delta}\right)}$$

Monotone decreasing w.r.t. VC dimension h $(h < n)$

- If samples are linear separable, empirical error is zero.

$$R_{\mathrm{emp}}[\widehat{f}] = 0$$

- The larger margin is, the smaller VC dim is.

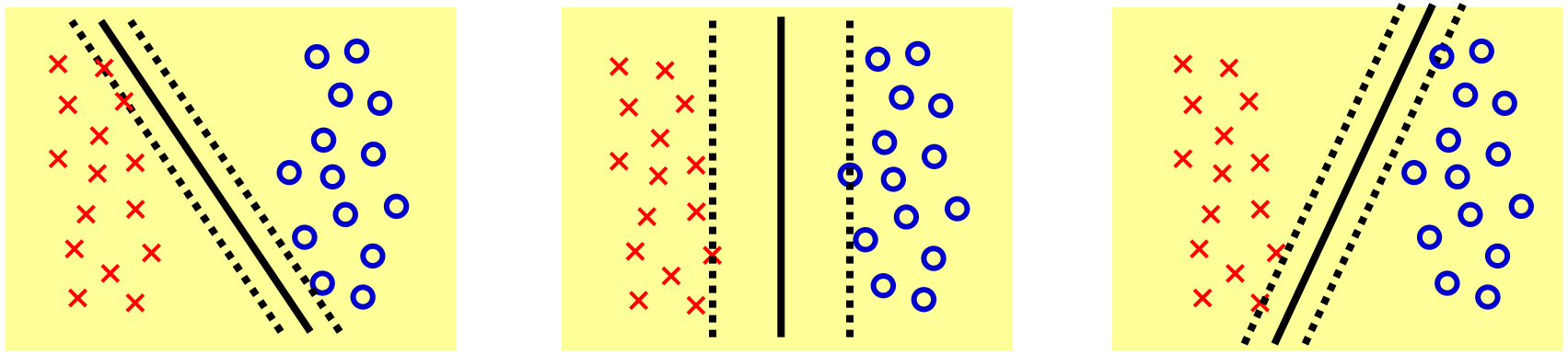


In VC theory, maximum margin classifier is optimal

Optimal Hyper-plane Classifier

- Separate two classes with maximum margin

Margin $1/\|\boldsymbol{w}\|$



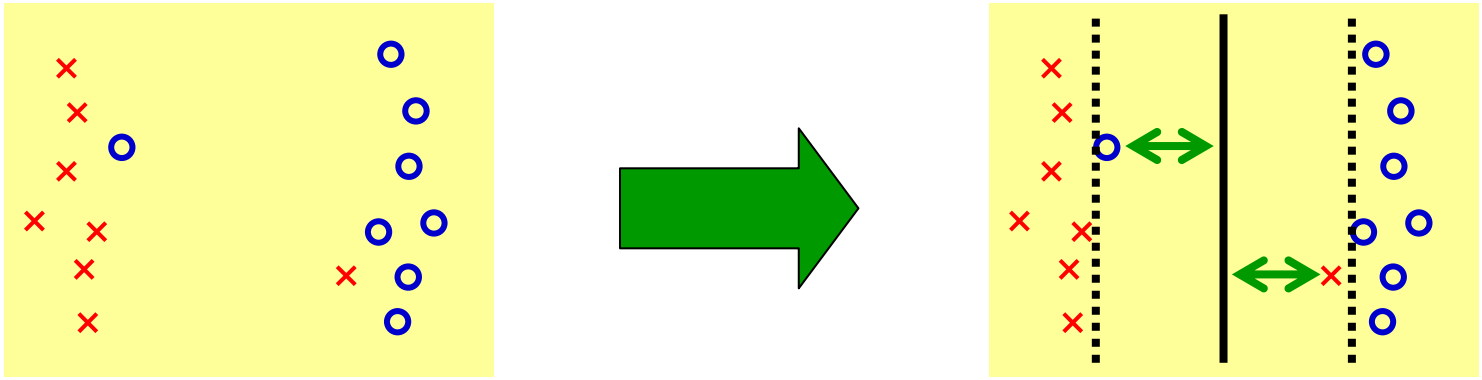
Small margin | Large margin | Small margin

$$\min_{\boldsymbol{w},b} \|\boldsymbol{w}\|^2$$

$$\text{subject to} \quad y_i \widehat{f}(\boldsymbol{x}_i) \geq 1 \quad \text{for } i = 1, \ldots, n.$$

Soft Margin

- If samples are not linearly separable, margin cannot be defined.
- Allow small error ξ_i.



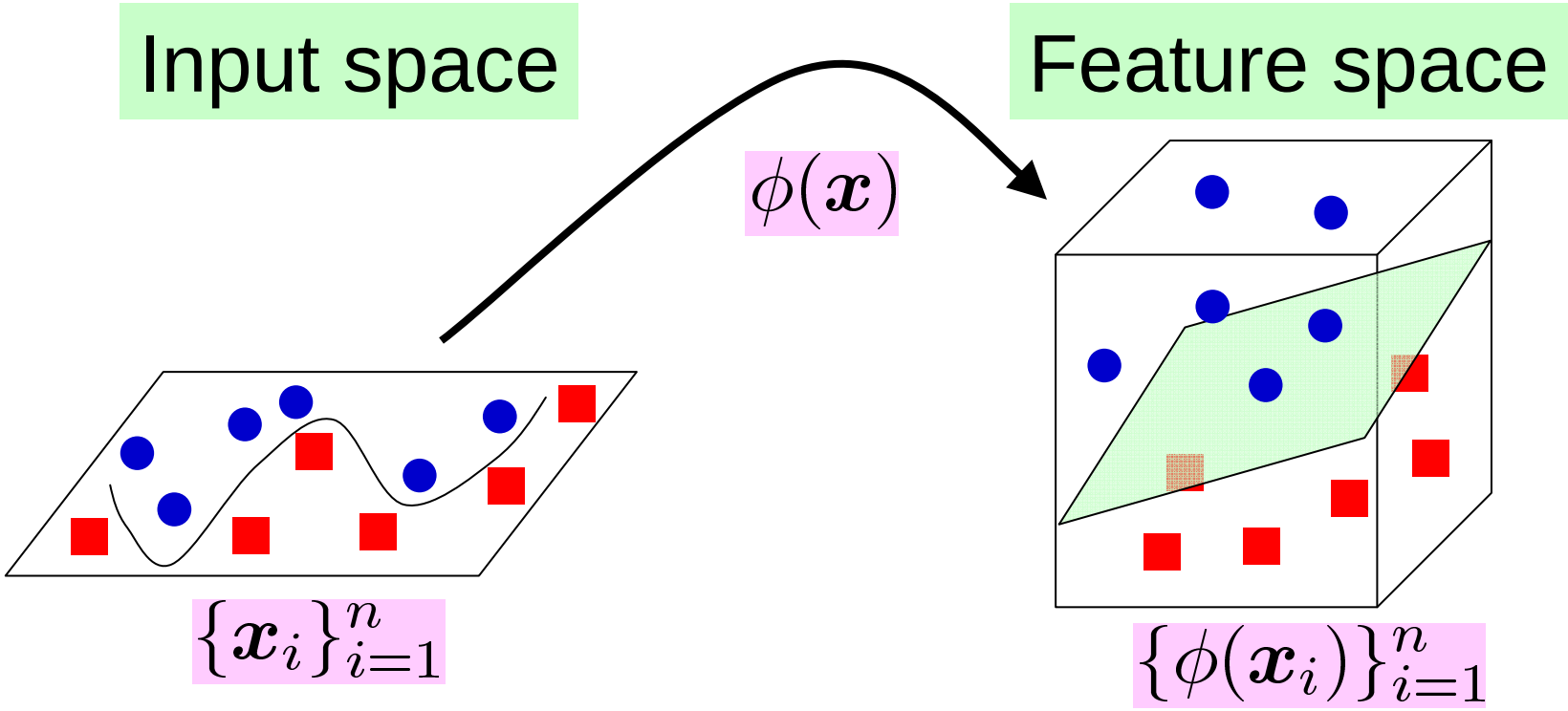
$$\min_{\boldsymbol{w},b} \|\boldsymbol{w}\|^2 + C\sum_{i=1}^{n} \xi_i$$

$$\text{subject to} \quad y_i \widehat{f}(\boldsymbol{x}_i) \geq 1 - \xi_i$$

$$\xi_i \geq 0 \quad \text{for } i = 1, \ldots, n.$$

Non-linear Extension

- Transform samples to a feature space by a non-linear mapping $\phi(\boldsymbol{x})$.
- Then find the maximum margin hyper-plane in the feature space.



Kernel Trick

- Compute inner product in the feature space by a kernel function:

$$\langle \phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}_j) \rangle = K(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

$$\forall \boldsymbol{x}, \boldsymbol{x}', \quad K(\boldsymbol{x}, \boldsymbol{x}') \geq 0$$

E.g., Gaussian kernel

$$K(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\|\boldsymbol{x} - \boldsymbol{x}'\|^2 / c^2\right)$$

- Any linear algorithm represented by inner product can be non-linearized by kernels
  - E.g.: Support vector machine, k-nearest neighbor classifier, principal component analysis, linear discriminant analysis, k-means clustering,

Notification of Final Assignment

1. Apply supervised learning techniques to your data set and analyze it.
2. Write your opinion about this course

- Final report deadline: Aug 6th (Fri.)
- E-mail submission is also accepted!

sugi@cs.titech.ac.jp

Mini-Workshop on Data Mining

- On July 20th (final class), we have a mini-workshop on data mining, instead of regular lecture.
- Several students present their data mining results.
- Those who give a talk at the workshop will have very good grades!

Mini-Workshop on Data Mining

- Application (just to declare that you want to give a presentation) deadline: June 29th.
- Presentation: 10-15(?) minutes.
  - Specification of your dataset
  - Employed methods
  - Outcome
- OHP or projector may be used.
- Slides should be in English.
- Better to speak in English, but Japanese is also allowed.

Schedule

- June 22nd : Preparation for workshop (no lecture)
- June 29th : Robust method (regular lecture)
- July 6th : Neural Networks (regular lecture)
- July 13th : Preparation for workshop (no lecture)
- July 20th : Mini-workshop
- July 27th : Mini-workshop (if necessary)

Homework

1. Prepare a toy binary classification problem (say 2-dim input) and test SVM. Then analyze the results by varying experimental conditions (datasets, kernels, regularization parameter C etc.).
   - Software is available from, e.g., http://www.support-vector.net/software.html
   - You may play with Java implementation, e.g., http://svm.dcs.rhbnc.ac.uk/pagesnew/GPat.shtml