서포트 벡터 학습을 이용한 함수근사
(Function Approximation via Support Vector Regression)

2004년 10월 30일
한국 퍼지 및 지능시스템 학회 2004년 추계학술대회 발표자료

고려대학교 과학기술대학 제어계측공학과
박 주 영

Korea University Jooyoung Park

Function Approximation via SVR October 2004

1. Introduction to support vector learning

- We may use the support vector learning algorithm to construct the following:
  - RBF networks
  - Two-layer perceptrons (i.e. with a single hidden layer)

- The support vectors consist of a small subset of the training data extracted by the algorithm.

2. Introduction to support vector regression(SVR)

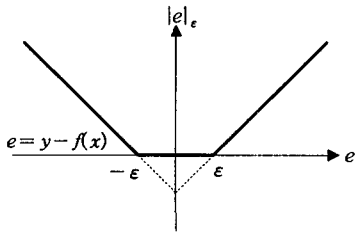
- Training data set:

  $\{(x_i, y_i)\}_{i=1}^{m}$, where x_i: the input pattern, y_i: target output

- Epsilon-insensitive loss function:

  $|e|_\varepsilon = \max(0, |e| - \varepsilon)$

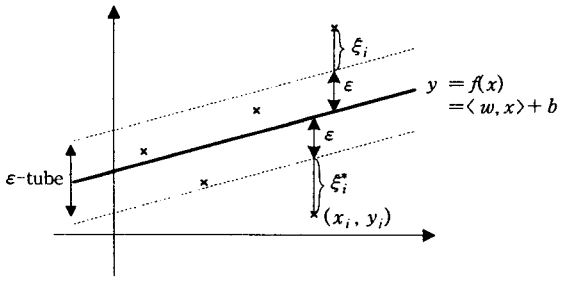
  $|y - f(x)|_\varepsilon = \max(0, |y - f(x)| - \varepsilon)$



- The SVR utilizes a linear approximator $f(x) = w^T x + b$, which deviates least from the training data according to the ϵ-insensitive loss function.

- More precisely, the objective of ε-SVR is to find the "flat" function $f(x) = \langle w, x \rangle + b$ while keeping the approximation error $|y_i - f(x_i)|_\varepsilon$ small.

- **Note**: This approximation problem can be formulated as an optimization problem, and its global min. can be found because
  ① its objective function is convex, and
  ② its constraints are linear in variables.



- Mathematical formulation for the SVR: Given the training data set $\{(x_i, y_i)\}_{i=1}^m$, solve the following quadratic problem:

$$\min \ \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m}(\xi_i + \xi_i^*)$$

$$s.t. \ \ y_i - (\langle w, x_i \rangle + b) \leq \xi_i + \varepsilon$$

$$(\langle w, x_i \rangle + b) - y_i \leq \varepsilon + \xi_i^*$$

$$\xi_i, \ \xi_i^* \geq 0 \ , \ \ i = 1, \cdots, m$$

3. A brief overview of optimization theory

- **Theorem**: $f \in C^1$ has a min. at $x^* \Rightarrow \frac{\partial f}{\partial x}(x^*) = 0$.

  This condition, together with convexity of f, is also a suff. cond.

- Example 1: $\min . f(x) = \frac{1}{2}(x_1^2 + x_2^2)$

  Solution:

$$\frac{\partial f}{\partial x} = 0 \ \Rightarrow \ \left[\frac{\partial f}{\partial x_1} \ \ \frac{\partial f}{\partial x_2}\right] = [x_1 \ x_2] = 0$$

$$\therefore \ x^* = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

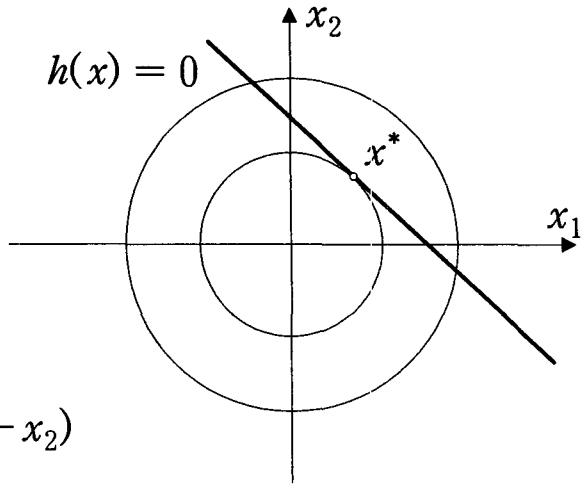
- In a constrained min. problem,

$$f \in C^1 \text{ has a min. at } x^* \not\Rightarrow \frac{\partial f}{\partial x}(x^*) = 0$$

- Example 2:

$$\min . f(x) = \frac{1}{2}(x_1^2 + x_2^2)$$

$$\text{s.t. } h(x) = 1 - x_1 - x_2 = 0$$



Solution: Define the Lagrange function

$$L(x, \lambda) = f(x) + \lambda h(x)$$
$$= \frac{1}{2}(x_1^2 + x_2^2) + \lambda(1 - x_1 - x_2)$$

$$\frac{\partial L}{\partial x} = 0 \Rightarrow [x_1 - \lambda \;\; x_2 - \lambda] = 0 . \;\therefore\; x_1 = x_2 = \lambda .$$

$$\frac{\partial L}{\partial \lambda} = 0 \Rightarrow 1 - x_1 - x_2 = 0. \;\therefore\; 1 - 2\lambda = 0. \;\therefore\; \lambda = \frac{1}{2} .$$

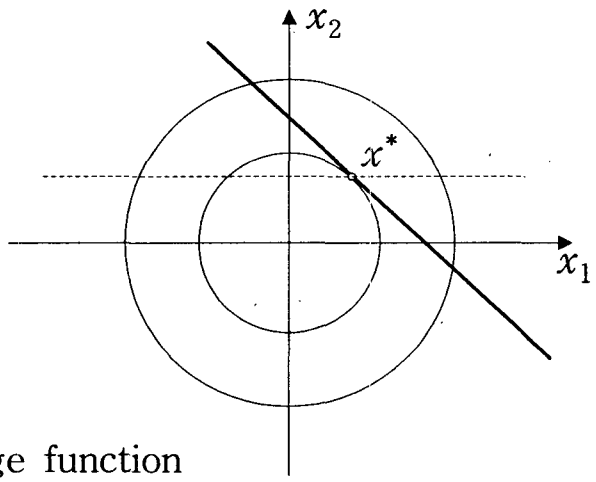
$$\therefore\; x^* = \begin{bmatrix} 1/2 \\ 1/2 \end{bmatrix}$$

- Example 3:

$$\min . f(x) = \frac{1}{2}(x_1^2 + x_2^2)$$

$$\text{s.t. } h(x) = 1 - x_1 - x_2 = 0 ,$$

$$g(x) = \frac{3}{4} - x_2 \le 0$$



Solution: Define the generalized Lagrange function

$$L(x, \lambda, \alpha) \triangleq f + \lambda h + \alpha g$$
$$= \frac{1}{2}(x_1^2 + x_2^2) + \lambda(1 - x_1 - x_2) + \alpha(\frac{3}{4} - x_2), \quad \alpha \ge 0$$

$$\frac{\partial L}{\partial x} = 0 \Rightarrow [x_1 - \lambda, x_2 - \lambda - \alpha] = 0. \;\therefore\; x_1 = \lambda , \; x_2 = \lambda + \alpha$$

$$\frac{\partial L}{\partial \lambda} = 0 \Rightarrow 1 - x_1 - x_2 = 0. \;\therefore\; 2\lambda + \alpha = 1$$

Also, $\alpha \geq 0$ and $\frac{3}{4} - x_2 \leq 0$.

One more condition is needed to solve the problem.

→ The Kuhn-Tucker complementarity condition

$$\alpha\left(\frac{3}{4} - x_2\right) = 0 \quad \text{i.e.,} \quad \alpha = 0 \ \text{ or } \ x_2 = \frac{3}{4}$$

① If $\alpha = 0$, then $\lambda = \frac{1}{2}$; thus $x_1 = x_2 = \frac{1}{2}$ ⊗ $(\because x_2 \geq \frac{3}{4})$

② If $x_2 = \frac{3}{4}$, then $\begin{cases} \lambda + \alpha = \frac{3}{4} \\ 2\lambda + \alpha = 1 \end{cases} \therefore \begin{cases} \lambda = \frac{1}{4}, \alpha = \frac{1}{2} \\ x_1 = \frac{1}{4}, x_2 = \frac{3}{4} \end{cases}$

$$\therefore x^* = \begin{bmatrix} 1/4 \\ 3/4 \end{bmatrix}$$

▸ **Theorem** (Kuhn-Tucker Theorem)

Given an opt. prob. with convex domain $\Omega \subseteq R^n$

$$\left.\begin{array}{ll} \min & f(x),\ x \in \Omega \ (x \text{ is primal variable}) \\ \text{s.t.} & g_i(x) \leq 0,\ i = 1, \cdots, k \\ & h_j(x) = 0,\ j = 1, \cdots, m \end{array}\right\} \text{ primal opt. prob } \bigstar$$

with $f \in C^1$ convex, and g_i, h_j affine, the following are necessary and sufficient condition for a point $x^* \in \Omega$ to be an opt.:

For $L(x, \alpha, \lambda) \triangleq f(x) + \sum_{i=1}^{k} \alpha_i g_i(x) + \sum_{j=1}^{m} \lambda_j h_j(x) = f + \alpha^T g + \lambda^T h$,

$\exists\, \alpha^*$ and h^* s.t. $\frac{\partial L}{\partial x}(x^*, \alpha^*, \lambda^*) = 0$, $\frac{\partial L}{\partial \lambda}(x^*, \alpha^*, \lambda^*) = 0$

$g_i(x^*) \leq 0$ and $\alpha_i^* \geq 0$ for $i = 1, \cdots, k$,

and $\underline{\alpha_i^* g_i(x^*) = 0,\ i = 1, \cdots, k}$

↳ The Kuhn-Tucker complementarity condition

▸ **Definition**: The (Lagrangian) dual prob. of ★ is the following:

$$\begin{cases} \max \ \Theta(\alpha, \lambda) & \text{(where } \alpha \text{ and } \lambda \text{ are dual variables)} \\ \text{s.t.} \quad \alpha \geq 0 \end{cases}$$

where $\Theta(\alpha, \lambda) \triangleq \inf_{x \in \Omega} L(x, \alpha, \lambda)$

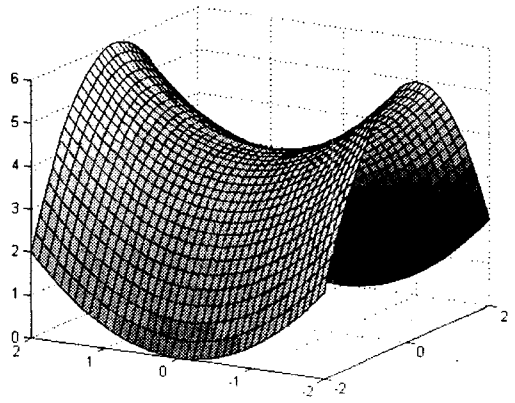
Dual problem of ★ :

$$\max \ \Theta(\alpha, \lambda)$$
$$\text{s.t.} \ \alpha \geq 0$$

▸ **Remarks**:

① In the course, our attention is focused on convex quadratic programs. (목적함수-Convex 2차식, 제약조건-1차식)

② It can be shown that for convex quadratic programs, L has a saddle point w.r.t. the primal and dual variables at the opt. soln.



4. Back to the SVR

- To find the best linear approximator for the training data set $\{(x_i, y_i)\}_{i=1}^m$, we need to solve the following problem:

$$\min \ \frac{1}{2}\|w\|^2 + C\sum_{i=1}^m(\xi_i + \xi_i^*)$$

$$s.t. \ \ y_i - (\langle w, x_i\rangle + b) \le \xi_i + \varepsilon,$$

$$(\langle w, x_i\rangle + b) - y_i \le \varepsilon + \xi_i^* \ , \ \ \xi_i, \ \xi_i^* \ge 0 \ , \ \ i = 1, \cdots, m$$

- Lagrange Function

$$L = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^m(\xi_i + \xi_i^*) + \sum_{i=1}^m \alpha_i[y_i - (\langle w, x_i\rangle + b) - \varepsilon - \xi_i]$$
$$+ \sum_{i=1}^m \alpha_i^*[(\langle w, x_i\rangle + b) - y_i - \varepsilon - \xi_i^*] - \sum_{i=1}^m(\eta_i\xi_i + \eta_i^*\xi_i^*)$$

$\alpha_i^{(*)}, \eta_i^{(*)} \ge 0$ / Primal variable: w, b, ξ, ξ^* / Dual variable: $\alpha, \alpha^*, \eta, \eta^*$

- The saddle point condition:

$$\frac{\partial L}{\partial w} = 0 \ \Leftrightarrow \ w - \sum_{i=1}^m \alpha_i x_i + \sum_{i=1}^m \alpha_i^* x_i = 0 \Leftrightarrow w = \sum_{i=1}^m(\alpha_i - \alpha_i^*)x_i$$

$$\frac{\partial L}{\partial b} = 0 \ \Leftrightarrow \ \sum_{i=1}^m(\alpha_i - \alpha_i^*) = 0$$

$$\frac{\partial L}{\partial \xi_i^{(*)}} = 0 \Leftrightarrow C - \alpha_i^{(*)} - \eta_i^{(*)} = 0 \Leftrightarrow \alpha_i^{(*)} + \eta_i^{(*)} = C \quad \therefore \alpha_i^{(*)} \in [0, C]$$

- Substitute the above into L to remove all primal variable $(w, b, \xi_i^{(*)})$
- Dual Problem:

$$\max \ D = -\frac{1}{2}\sum_{i=1}^m\sum_{j=1}^m(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)\langle x_i, x_j\rangle$$
$$+ \sum_{i=1}^m(\alpha_i - \alpha_i^*)y_i - \sum_{i=1}^m(\alpha_i + \alpha_j^*)\varepsilon$$

$$s.t. \ \ \sum_{i=1}^m(\alpha_i - \alpha_i^*) = 0 \ , \ \ \alpha_i^{(*)} \in [0, C], \ i = 1, \cdots, m$$

- $w = \sum_{i=1}^{m}(\alpha_i - \alpha_i^*)x_i$

$$\therefore\ f(x) = \langle w, x\rangle + b = \sum_{i=1}^{m}(\alpha_i - \alpha_i^*)\langle x_i, x\rangle + b$$

- The KT condition
  - ① $\alpha_i[y_i - (\langle w, x_i\rangle + b) - \varepsilon - \xi_i] = 0,\ \forall i$
  - ② $\alpha_i^*[(\langle w, x_i\rangle + b) - y_i - \varepsilon - \xi_i^*] = 0,\ \forall i$
  - ③ $\eta_i\xi_i = 0$ *i.e.*, $(C - \alpha_i)\xi_i = 0,\ \forall i$
  - ④ $\eta_i^*\xi_i^* = 0$ *i.e.*, $(C - \alpha_i^*)\xi_i^* = 0,\ \forall i$

- Remark:
  - ① $\alpha_i\alpha_i^* = 0 \quad \because \alpha_i \neq 0 \Rightarrow y_i - (\langle w, x_i\rangle + b) - \varepsilon - \xi_i = 0$

$$\Rightarrow y_i - (\langle w, x_i\rangle + b) = \varepsilon + \xi_i$$

$$\Rightarrow \underbrace{(\langle w, x_i\rangle + b) - y_i - \varepsilon - \xi_i}_{= -\varepsilon - \xi_i} < 0$$

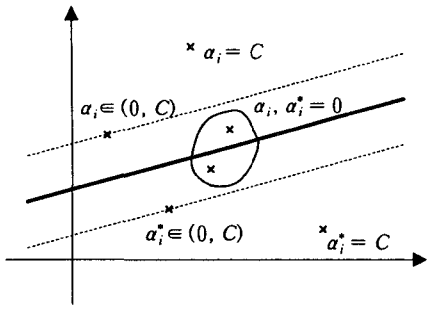
$$\Rightarrow \alpha_i^* = 0$$

Similarly for the other way

- ② $\alpha_i \in (0, C) \Rightarrow \xi_i = 0$ and $y_i - (\langle w, x_i\rangle + b) = \varepsilon$
- ③ $\alpha_i^* \in (0, C) \Rightarrow \xi_i^* = 0$ and $(\langle w, x_i\rangle + b) - y_i = \varepsilon$
- ④ $\xi_i > 0 \Rightarrow \alpha_i = C$
- ⑤ $\xi_i^* > 0 \Rightarrow \alpha_i^* = C$

⑥ ε-tube 내부에 training data에 대응하는 $\alpha_i, \alpha_i^* = 0$.

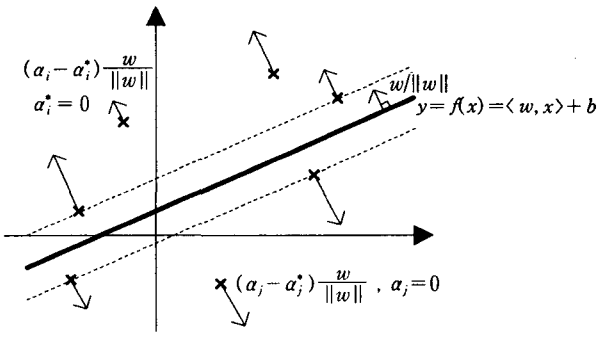
(∵ the KT condition ①,② 모두 두 번째항이 nonzero 이므로)



▸ Note:

$$\text{목적함수 } D = -\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)\langle x_i, x_j\rangle + \sum_{i=1}^{m}(\alpha_i - \alpha_i^*)y_i - \sum_{i=1}^{m}(\alpha_i + \alpha_j^*)\varepsilon$$

▸ Mechanical Interpretation

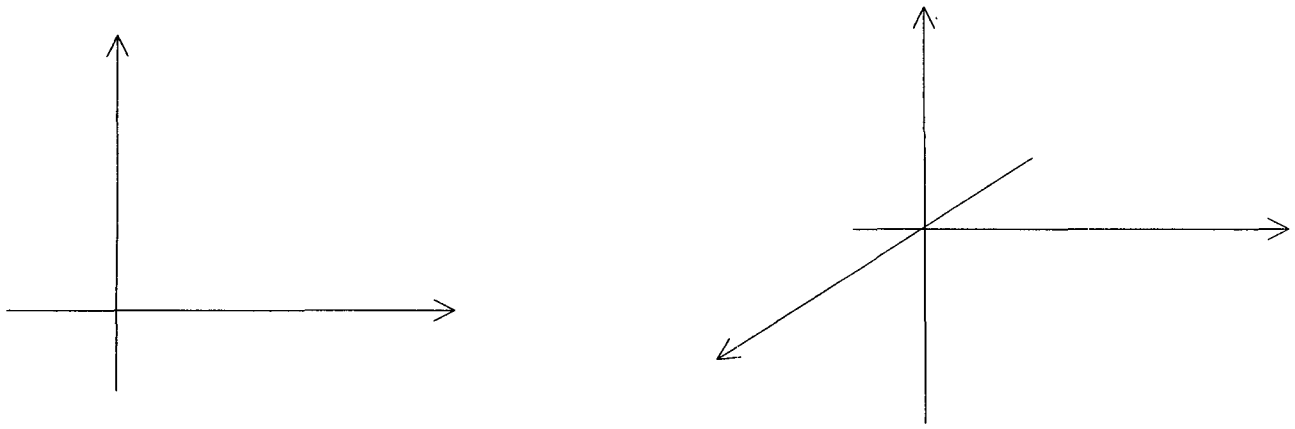


$\sum_{i=1}^{m}(\alpha_i - \alpha_i^*) = 0$: Force balance(힘 균형)

$w = \sum_{i=1}^{m}(\alpha_i - \alpha_i^*)x_i$,

$\therefore \sum_{i=1}^{m} x_i \times (\alpha_i - \alpha_i^*)\frac{w}{\|w\|} = w \times \frac{w}{\|w\|} = 0$: Torque balance(회전 모멘트 균형)

- An extension for nonlinear approximators:



- Feature space : Given training data $\{(x_i, y_i)\}_{i=1}^{m}$,

  where $x_i \in R^l$, $y_i \in R$, preprocess the data with

$$\phi: \begin{array}{ccc} R^l & \to & F \\ x & \mapsto & \phi(x), \end{array} \quad \text{where } l \ll \dim(F),$$

  to get $(\phi(x_i), y_i) \in F \times R$, $i = 1, \cdots, m$.

- Example:

$$\phi: \begin{array}{ccc} R^2 & \to & R^3 \\ (x_1, x_2) & \mapsto & (x_1^2, \sqrt{2}x_1x_2, x_2^2) \end{array}$$

- The kernel trick

$$\begin{aligned} \langle \phi(x), \phi(y) \rangle &= (x_1^2, \sqrt{2}x_1x_2, x_2^2) \cdot (y_1^2, \sqrt{2}y_1y_2, y_2^2) \\ &= x_1^2y_1^2 + 2x_1x_2y_1y_2 + x_2^2y_2^2 \\ &= (x_1y_1 + x_2y_2)^2 \\ &= (\langle x, y \rangle)^2 \\ &= K(x, y) \end{aligned}$$

  Note : $\langle \phi(x), \phi(y) \rangle = (\langle x, y \rangle)^2 = K(x, y)$ can be computed easily on the input space !!!

- Question: Which kernel function $K(x, y)$ has a corresponding inner product $\langle \phi(x), \phi(y) \rangle$ in some feature space?
  Answer: Mercer kernels.

- **Mercer's Theorem** (1909): Roughly speaking,
  if $\int \int K(x, y) f(x) f(y) dx dy \geq 0$ for $\forall f \in L_2$,
  then $\exists \phi: R^l \rightarrow F$ s.t. $K(x, y) = \langle \phi(x), \phi(y) \rangle$.
- Examples of Mercer kernels:
  - RBF: $K(x, y) = \exp(-\frac{1}{2}\frac{\|x-y\|^2}{\sigma^2})$
  - MLP: $K(x, y) = \tanh(\gamma\langle x, y\rangle + \theta)$, where $\gamma, \theta > 0$
  - Polynomial: $K(x, y) = (\langle x, y\rangle + c)^d$
- Note: Recall the note of p. 17.

- Kernel-based nonlinear approximators:
  If we use the nonlinear approximator $f(x) = \langle w, \phi(x) \rangle + b$, we have

$$w = \sum_{i=1}^{m} (\alpha_i - \alpha_i^*) \phi(x_i),$$

$$f(x) = \langle w, \phi(x) \rangle + b = \sum_{i=1}^{m} (\alpha_i - \alpha_i^*) k(x_i, x) + b$$

$\therefore$ Dual problem

$$\Leftrightarrow \max_{\alpha_i, \alpha_i^*} -\frac{1}{2}\sum_{i}^{m}\sum_{j}^{m}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)\, \boldsymbol{k(x_i, x_j)} + \sum_{i}^{m}(\alpha_i - \alpha_i^*) y_i - \sum_{i}^{m}(\alpha_i + \alpha_j^*)\varepsilon$$

$$s.t. \sum_{i}^{m}(\alpha_i - \alpha_i^*) = 0, \quad \alpha_i^{(*)} \in [0, C], \quad i = 1, \cdots, m$$

$$\Leftrightarrow \quad \min_{\underline{\alpha}_i,\ \underline{\alpha}_i^*} \ \frac{1}{2}(\underline{\alpha}-\underline{\alpha}^*)^T \boldsymbol{K}(\underline{\alpha}-\underline{\alpha}^*)+\begin{bmatrix}\varepsilon\underline{1}\\ \varepsilon\underline{1}\end{bmatrix}^T\begin{bmatrix}\underline{\alpha}\\ \underline{\alpha}^*\end{bmatrix}-\begin{bmatrix}\underline{y}\\ -\underline{y}\end{bmatrix}^T\begin{bmatrix}\underline{\alpha}\\ \underline{\alpha}^*\end{bmatrix}$$

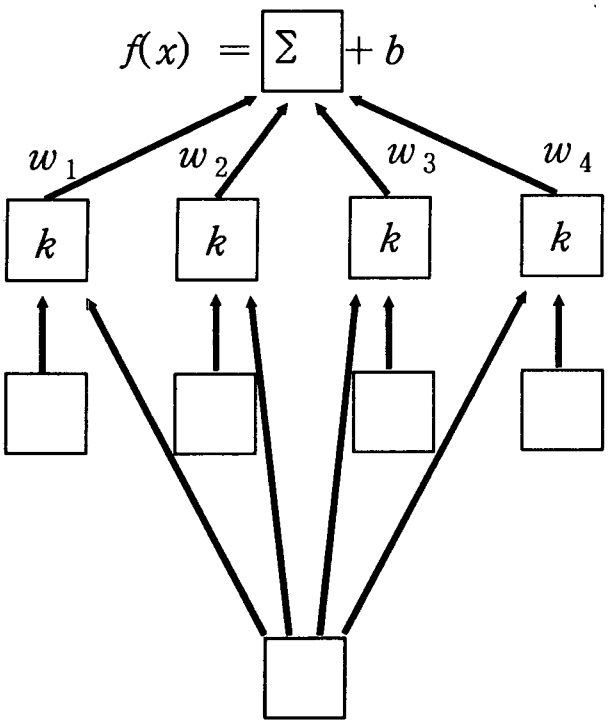
$$s.t.\begin{bmatrix}\underline{1}\\ -\underline{1}\end{bmatrix}^T\begin{bmatrix}\underline{\alpha}\\ \underline{\alpha}^*\end{bmatrix}=0\ ,\ \begin{bmatrix}0\\ \vdots\\ 0\end{bmatrix}\le\begin{bmatrix}\underline{\alpha}\\ \underline{\alpha}^*\end{bmatrix}\le\begin{bmatrix}C\\ \vdots\\ C\end{bmatrix}$$

$$\Leftrightarrow \quad \min_{\underline{\alpha}_i,\ \underline{\alpha}_i^*} \ \frac{1}{2}\begin{bmatrix}\underline{\alpha}\\ \underline{\alpha}^*\end{bmatrix}^T\begin{bmatrix}K & -K\\ -K & K\end{bmatrix}\begin{bmatrix}\underline{\alpha}\\ \underline{\alpha}^*\end{bmatrix}+\begin{bmatrix}\varepsilon\underline{1}-\underline{y}\\ \varepsilon\underline{1}+\underline{y}\end{bmatrix}^T\begin{bmatrix}\underline{\alpha}\\ \underline{\alpha}^*\end{bmatrix}$$

$$s.t.\begin{bmatrix}\underline{1}\\ -\underline{1}\end{bmatrix}^T\begin{bmatrix}\underline{\alpha}\\ \underline{\alpha}^*\end{bmatrix}=0\ ,\ \begin{bmatrix}0\\ \vdots\\ 0\end{bmatrix}\le\begin{bmatrix}\underline{\alpha}\\ \underline{\alpha}^*\end{bmatrix}\le\begin{bmatrix}C\\ \vdots\\ C\end{bmatrix}$$

(MATLAB의 "QP solver quadprog"에 적합한 꼴)

5. The SVR architecture



approximator $f(x)=\sum w_i k(x,x_i)+b$

hidden nodes $k(x,x_i)$

support vectors $x_1,\ldots,x_4$

input vector x

6. Discussion

Given the data set $\{(x_i, y_i)\}_{i=1}^{m}$, we have the following kinds of training problems:

- Least square problem: w_i?

$$f(x) = \sum_{i=1}^{m_1} w_i g(x; \mu_i, \sigma_i), \quad \text{where } m_1, \mu_i, \sigma_i \text{ are fixed}$$

- Learning via back-propagation: w_i, μ_i, σ_i?

$$f(x) = \sum_{i=1}^{m_2} w_i g(x; \mu_i, \sigma_i), \quad \text{where } m_2 \text{ is fixed}$$

- Support vector learning: w_i?

$$f(x) = \sum_{i=1}^{m} w_i g(x; x_i, \sigma_i), \text{ where } \sigma_i \text{ is fixed and } \#(\text{nonzero } w_i) << m$$

7. References

[1] Bernhard Schölkopf, Alexander J. Smola, *Learning with Kernels*, MIT Press, 2002.

[2] Nello Cristianini, John Shawe-Taylor, *An Introduction to Support Vector Machines*, Cambridge University Press, 2000.

[3] http://www.kernel-machines.org

[4] http://www.isis.ecs.soton.ac.kr/resources/svminfo/