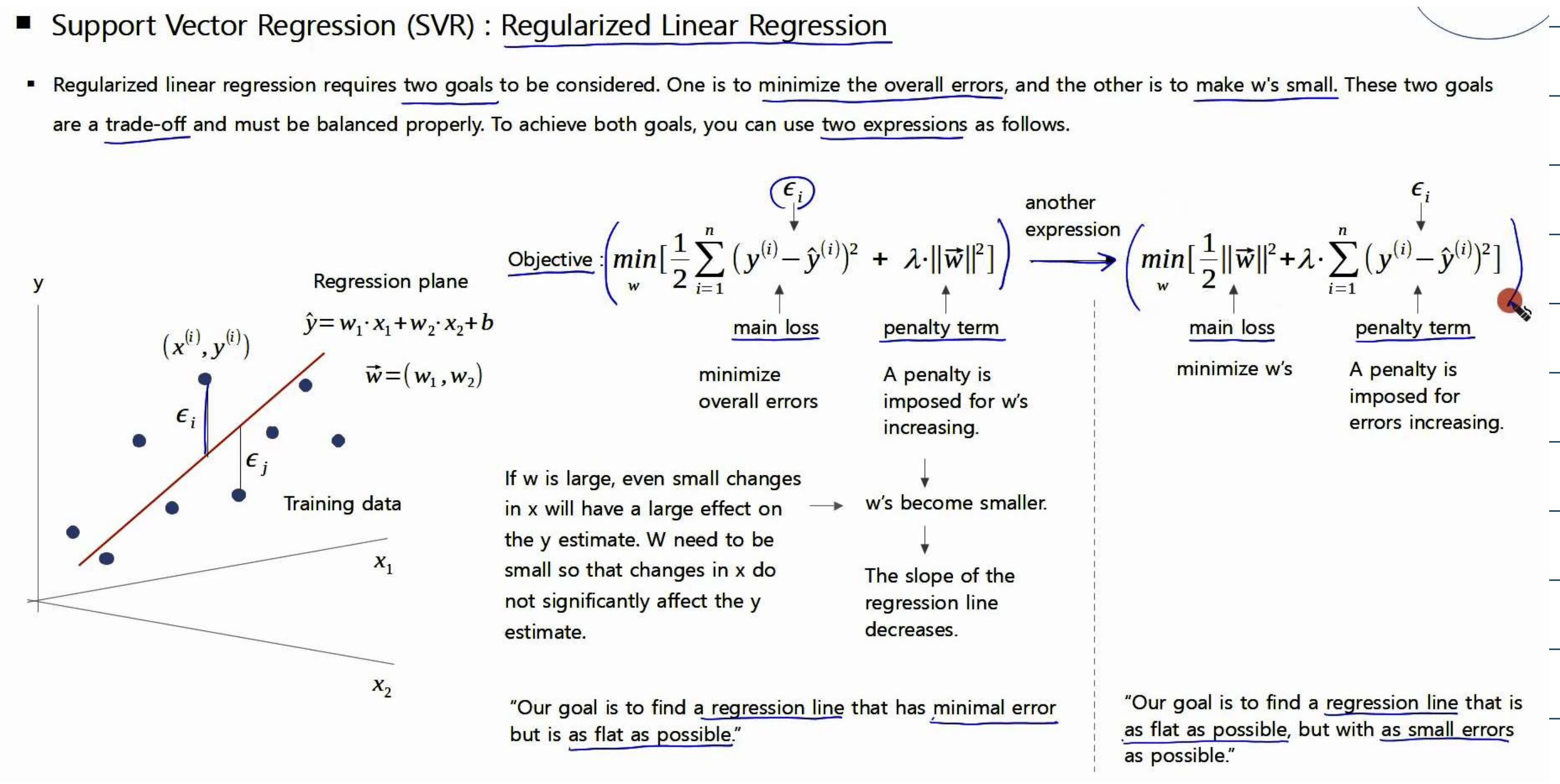
Support Vector Regression (SVR)

Friday 24 May 2024 1:19 AM



Support Vector Regression (SVR) : Regularized Linear Regression

- Regularized linear regression requires two goals to be considered. One is to minimize the overall errors, and the other is to make w's small. These two goals are a trade-off and must be balanced properly. To achieve both goals, you can use two expressions as follows.

ϵ_i

Objective : $\left(\min_w[\frac{1}{2}\sum_{i=1}^{n}(y^{(i)}-\hat{y}^{(i)})^2 + \lambda\cdot\|\vec{w}\|^2]\right) \xrightarrow{\text{another expression}} \left(\min_w[\frac{1}{2}\|\vec{w}\|^2+\lambda\cdot\sum_{i=1}^{n}(y^{(i)}-\hat{y}^{(i)})^2]\right)$

ϵ_i

main loss — minimize overall errors

penalty term — A penalty is imposed for w's increasing.

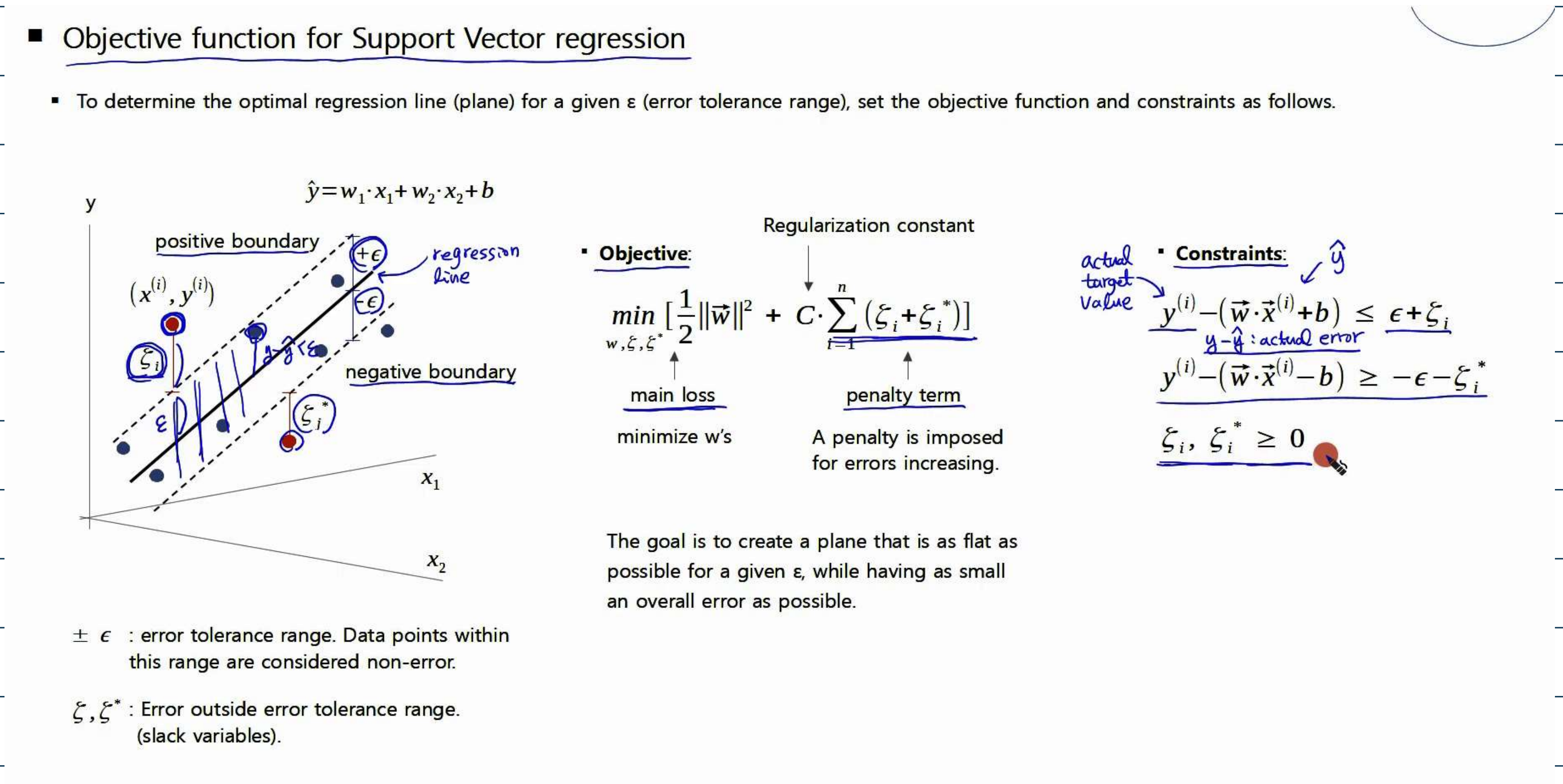
If w is large, even small changes in x will have a large effect on the y estimate. W need to be small so that changes in x do not significantly affect the y estimate. → w's become smaller. ↓ The slope of the regression line decreases.

"Our goal is to find a regression line that has minimal error but is as flat as possible."

main loss — minimize w's

penalty term — A penalty is imposed for errors increasing.

"Our goal is to find a regression line that is as flat as possible, but with as small errors as possible."



Objective function for Support Vector regression

- To determine the optimal regression line (plane) for a given ε (error tolerance range), set the objective function and constraints as follows.

$\pm\ \epsilon$: error tolerance range. Data points within this range are considered non-error.

ζ,ζ^* : Error outside error tolerance range. (slack variables).

- **Objective**:

Regularization constant

$$\min_{w,\zeta,\zeta^*}[\frac{1}{2}\|\vec{w}\|^2 + C\cdot\sum_{i=1}^{n}(\zeta_i+\zeta_i^*)]$$

main loss — minimize w's

penalty term — A penalty is imposed for errors increasing.

The goal is to create a plane that is as flat as possible for a given ε, while having as small an overall error as possible.

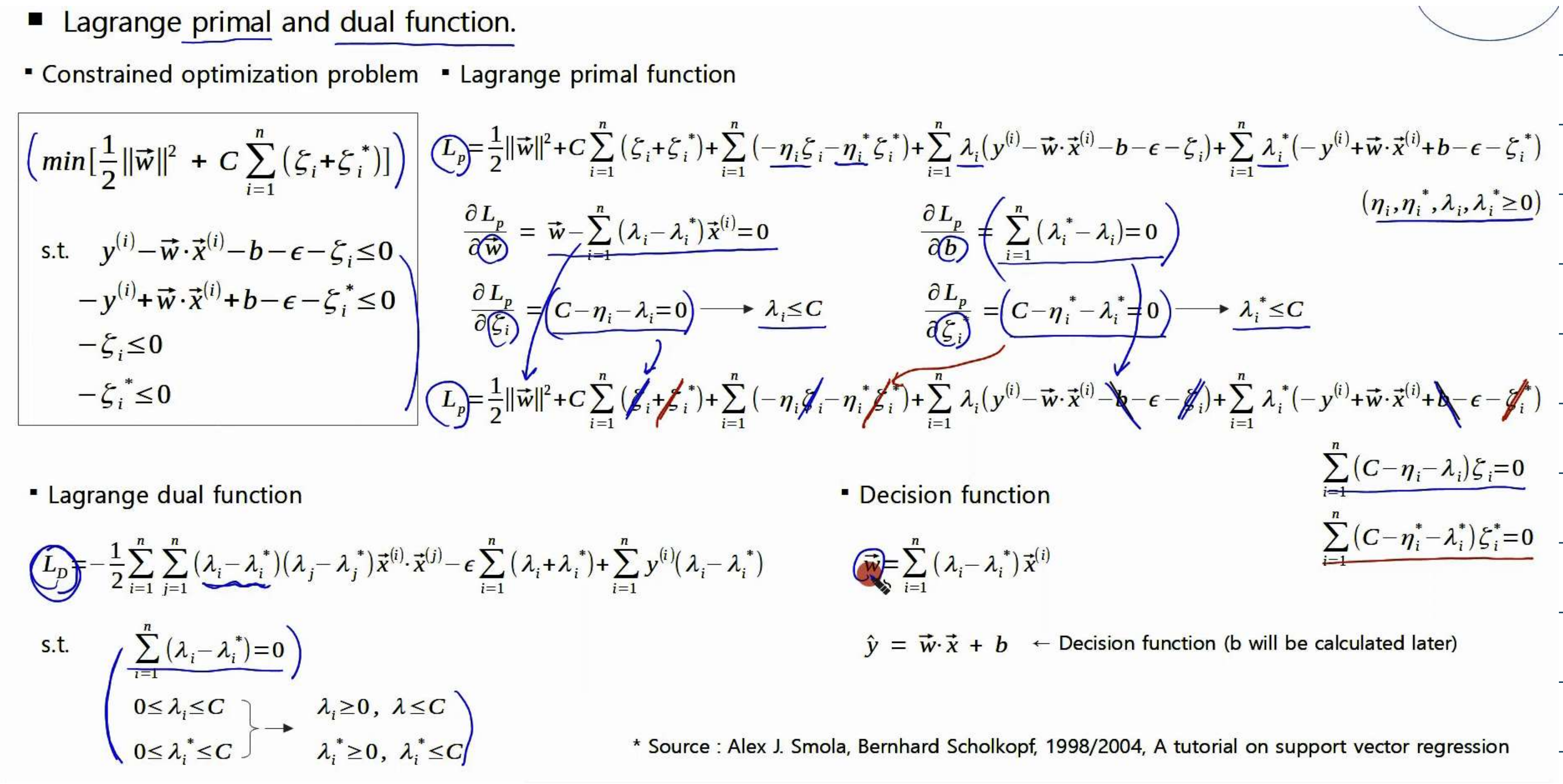
- **Constraints**:

actual target value

$$\underbrace{y^{(i)}-(\overset{\hat{y}}{\vec{w}\cdot\vec{x}^{(i)}+b})}_{y-\hat{y}\text{ : actual error}} \le \epsilon+\zeta_i$$

$$y^{(i)}-(\vec{w}\cdot\vec{x}^{(i)}-b) \ge -\epsilon-\zeta_i^*$$

$$\zeta_i,\ \zeta_i^* \ge 0$$



Lagrange primal and dual function.

- Constrained optimization problem

$$\left(\min\left[\frac{1}{2}\|\vec{w}\|^2 + C\sum_{i=1}^{n}(\zeta_i+\zeta_i^*)\right]\right)$$

s.t. $y^{(i)}-\vec{w}\cdot\vec{x}^{(i)}-b-\epsilon-\zeta_i\le 0$

$-y^{(i)}+\vec{w}\cdot\vec{x}^{(i)}+b-\epsilon-\zeta_i^*\le 0$

$-\zeta_i\le 0$

$-\zeta_i^*\le 0$

- Lagrange primal function

$$L_p=\frac{1}{2}\|\vec{w}\|^2+C\sum_{i=1}^{n}(\zeta_i+\zeta_i^*)+\sum_{i=1}^{n}(-\underline{\eta_i}\zeta_i-\underline{\eta_i^*}\zeta_i^*)+\sum_{i=1}^{n}\underline{\lambda_i}(y^{(i)}-\vec{w}\cdot\vec{x}^{(i)}-b-\epsilon-\zeta_i)+\sum_{i=1}^{n}\underline{\lambda_i^*}(-y^{(i)}+\vec{w}\cdot\vec{x}^{(i)}+b-\epsilon-\zeta_i^*)$$

$\underline{(\eta_i,\eta_i^*,\lambda_i,\lambda_i^*\ge 0)}$

$$\frac{\partial L_p}{\partial \vec{w}}=\underline{\vec{w}-\sum_{i=1}^{n}(\lambda_i-\lambda_i^*)\vec{x}^{(i)}=0}\qquad \frac{\partial L_p}{\partial b}=\underline{\sum_{i=1}^{n}(\lambda_i^*-\lambda_i)=0}$$

$$\frac{\partial L_p}{\partial \zeta_i}=(C-\eta_i-\lambda_i=0)\longrightarrow \underline{\lambda_i\le C}\qquad \frac{\partial L_p}{\partial \zeta_i^*}=(C-\eta_i^*-\lambda_i^*=0)\longrightarrow \underline{\lambda_i^*\le C}$$

$$L_p=\frac{1}{2}\|\vec{w}\|^2+C\sum_{i=1}^{n}(\cancel{\zeta_i}+\cancel{\zeta_i^*})+\sum_{i=1}^{n}(-\eta_i\cancel{\zeta_i}-\eta_i^*\cancel{\zeta_i^*})+\sum_{i=1}^{n}\lambda_i(y^{(i)}-\vec{w}\cdot\vec{x}^{(i)}-\cancel{b}-\epsilon-\cancel{\zeta_i})+\sum_{i=1}^{n}\lambda_i^*(-y^{(i)}+\vec{w}\cdot\vec{x}^{(i)}+\cancel{b}-\epsilon-\cancel{\zeta_i^*})$$

$\underline{\sum_{i=1}^{n}(C-\eta_i-\lambda_i)\zeta_i=0}$

$\underline{\sum_{i=1}^{n}(C-\eta_i^*-\lambda_i^*)\zeta_i^*=0}$

- Lagrange dual function

$$L_D=-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}(\lambda_i-\lambda_i^*)(\lambda_j-\lambda_j^*)\vec{x}^{(i)}\cdot\vec{x}^{(j)}-\epsilon\sum_{i=1}^{n}(\lambda_i+\lambda_i^*)+\sum_{i=1}^{n}y^{(i)}(\lambda_i-\lambda_i^*)$$

s.t. $\underline{\sum_{i=1}^{n}(\lambda_i-\lambda_i^*)=0}$

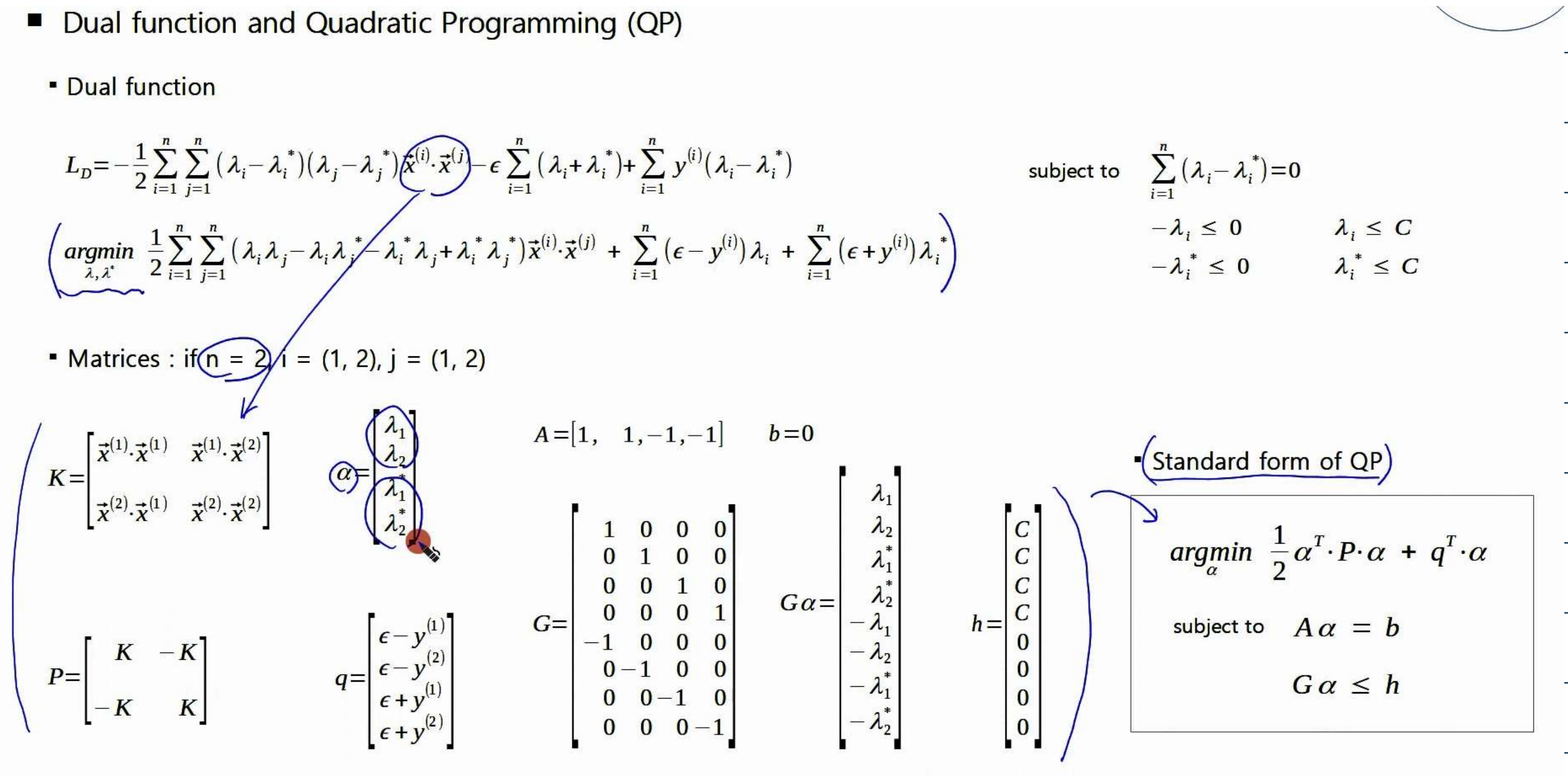
$$\left.\begin{array}{l}0\le\lambda_i\le C\\ 0\le\lambda_i^*\le C\end{array}\right\}\rightarrow\begin{array}{l}\lambda_i\ge 0,\ \lambda\le C\\ \lambda_i^*\ge 0,\ \lambda_i^*\le C\end{array}$$

- Decision function

$$\vec{w}=\sum_{i=1}^{n}(\lambda_i-\lambda_i^*)\vec{x}^{(i)}$$

$\hat{y}=\vec{w}\cdot\vec{x}+b$ ← Decision function (b will be calculated later)

* Source : Alex J. Smola, Bernhard Scholkopf, 1998/2004, A tutorial on support vector regression



Dual function and Quadratic Programming (QP)

- Dual function

$$L_D=-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}(\lambda_i-\lambda_i^*)(\lambda_j-\lambda_j^*)\vec{x}^{(i)}\cdot\vec{x}^{(j)}-\epsilon\sum_{i=1}^{n}(\lambda_i+\lambda_i^*)+\sum_{i=1}^{n}y^{(i)}(\lambda_i-\lambda_i^*)$$

$$\left(\underset{\lambda,\lambda^*}{\operatorname{argmin}}\ \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}(\lambda_i\lambda_j-\lambda_i\lambda_j^*-\lambda_i^*\lambda_j+\lambda_i^*\lambda_j^*)\vec{x}^{(i)}\cdot\vec{x}^{(j)}+\sum_{i=1}^{n}(\epsilon-y^{(i)})\lambda_i+\sum_{i=1}^{n}(\epsilon+y^{(i)})\lambda_i^*\right)$$

subject to $\sum_{i=1}^{n}(\lambda_i-\lambda_i^*)=0$

$-\lambda_i\le 0\qquad \lambda_i\le C$

$-\lambda_i^*\le 0\qquad \lambda_i^*\le C$

- Matrices : if n = 2, i = (1, 2), j = (1, 2)

$$K=\begin{bmatrix}\vec{x}^{(1)}\cdot\vec{x}^{(1)} & \vec{x}^{(1)}\cdot\vec{x}^{(2)}\\ \vec{x}^{(2)}\cdot\vec{x}^{(1)} & \vec{x}^{(2)}\cdot\vec{x}^{(2)}\end{bmatrix}\qquad \alpha=\begin{bmatrix}\lambda_1\\ \lambda_2\\ \lambda_1^*\\ \lambda_2^*\end{bmatrix}\qquad A=[1,\ \ 1,-1,-1]\qquad b=0$$

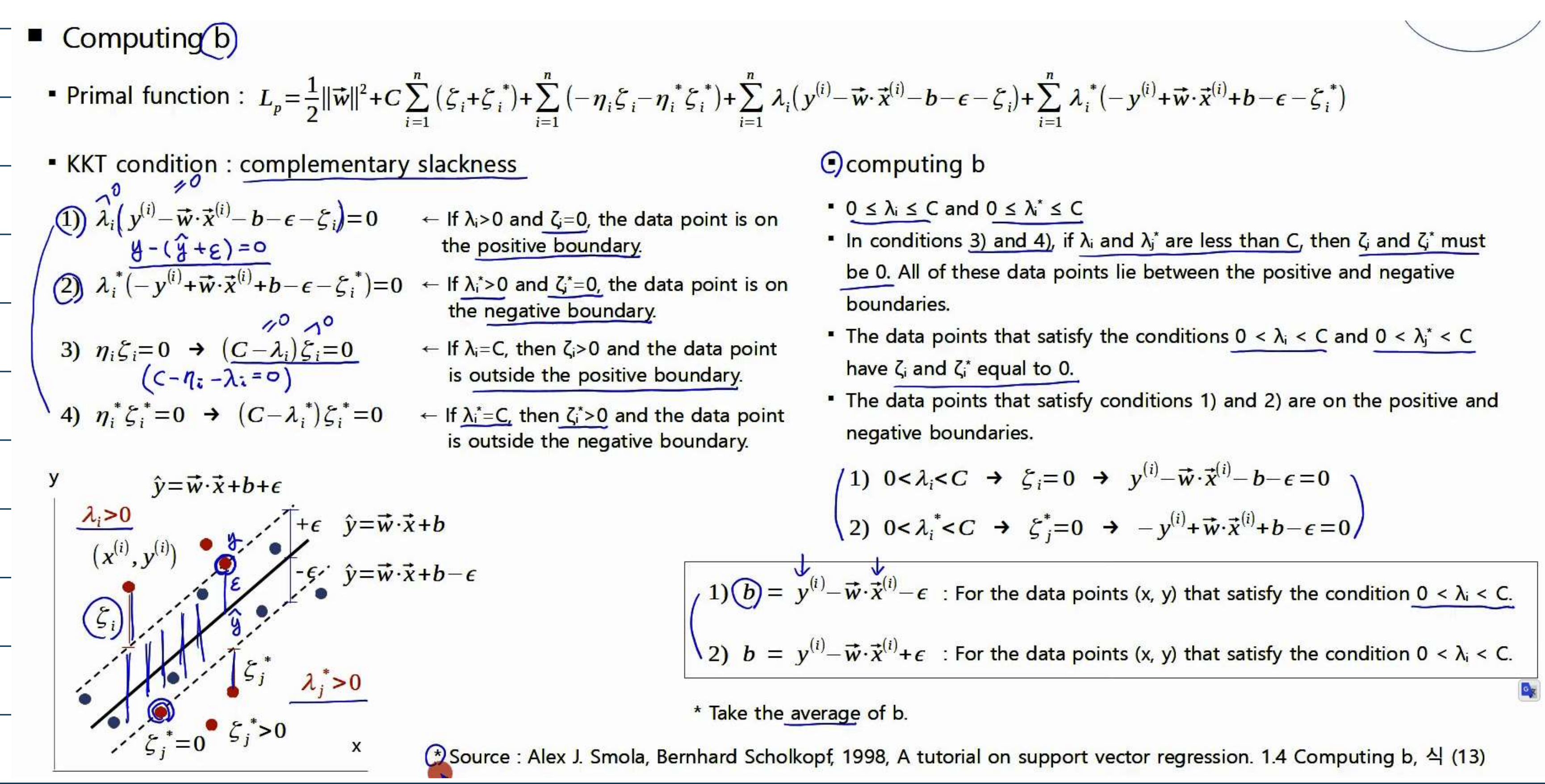
$$P=\begin{bmatrix}K & -K\\ -K & K\end{bmatrix}\qquad q=\begin{bmatrix}\epsilon-y^{(1)}\\ \epsilon-y^{(2)}\\ \epsilon+y^{(1)}\\ \epsilon+y^{(2)}\end{bmatrix}\qquad G=\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\\-1&0&0&0\\0&-1&0&0\\0&0&-1&0\\0&0&0&-1\end{bmatrix}\qquad G\alpha=\begin{bmatrix}\lambda_1\\ \lambda_2\\ \lambda_1^*\\ \lambda_2^*\\ -\lambda_1\\ -\lambda_2\\ -\lambda_1^*\\ -\lambda_2^*\end{bmatrix}\qquad h=\begin{bmatrix}C\\C\\C\\C\\0\\0\\0\\0\end{bmatrix}$$

- Standard form of QP

$$\underset{\alpha}{\operatorname{argmin}}\ \frac{1}{2}\alpha^T\cdot P\cdot\alpha+q^T\cdot\alpha$$

subject to $A\alpha=b$

$G\alpha\le h$



Computing b

- Primal function : $L_p=\frac{1}{2}\|\vec{w}\|^2+C\sum_{i=1}^{n}(\zeta_i+\zeta_i^*)+\sum_{i=1}^{n}(-\eta_i\zeta_i-\eta_i^*\zeta_i^*)+\sum_{i=1}^{n}\lambda_i(y^{(i)}-\vec{w}\cdot\vec{x}^{(i)}-b-\epsilon-\zeta_i)+\sum_{i=1}^{n}\lambda_i^*(-y^{(i)}+\vec{w}\cdot\vec{x}^{(i)}+b-\epsilon-\zeta_i^*)$

- KKT condition : complementary slackness

1) $\overset{>0}{\lambda_i}\left(\overset{\neq 0}{y^{(i)}-\vec{w}\cdot\vec{x}^{(i)}-b-\epsilon-\zeta_i}\right)=0$ ← If λᵢ>0 and ζᵢ=0, the data point is on the positive boundary.

$\underline{y-(\hat{y}+\varepsilon)=0}$

2) $\lambda_i^*(-y^{(i)}+\vec{w}\cdot\vec{x}^{(i)}+b-\epsilon-\zeta_i^*)=0$ ← If λᵢ*>0 and ζᵢ*=0, the data point is on the negative boundary.

3) $\eta_i\zeta_i=0\ \rightarrow\ \underline{(\overset{\neq 0}{C-\lambda_i})\overset{>0}{\zeta_i}=0}$ ← If λᵢ=C, then ζᵢ>0 and the data point is outside the positive boundary.

$(C-\eta_i-\lambda_i=0)$

4) $\eta_i^*\zeta_i^*=0\ \rightarrow\ (C-\lambda_i^*)\zeta_i^*=0$ ← If λᵢ*=C, then ζᵢ*>0 and the data point is outside the negative boundary.

- computing b

- 0 ≤ λᵢ ≤ C and 0 ≤ λᵢ* ≤ C
- In conditions 3) and 4), if λᵢ and λⱼ* are less than C, then ζᵢ and ζᵢ* must be 0. All of these data points lie between the positive and negative boundaries.
- The data points that satisfy the conditions 0 < λᵢ < C and 0 < λⱼ* < C have ζᵢ and ζᵢ* equal to 0.
- The data points that satisfy conditions 1) and 2) are on the positive and negative boundaries.

1) $0<\lambda_i<C\ \rightarrow\ \zeta_i=0\ \rightarrow\ y^{(i)}-\vec{w}\cdot\vec{x}^{(i)}-b-\epsilon=0$

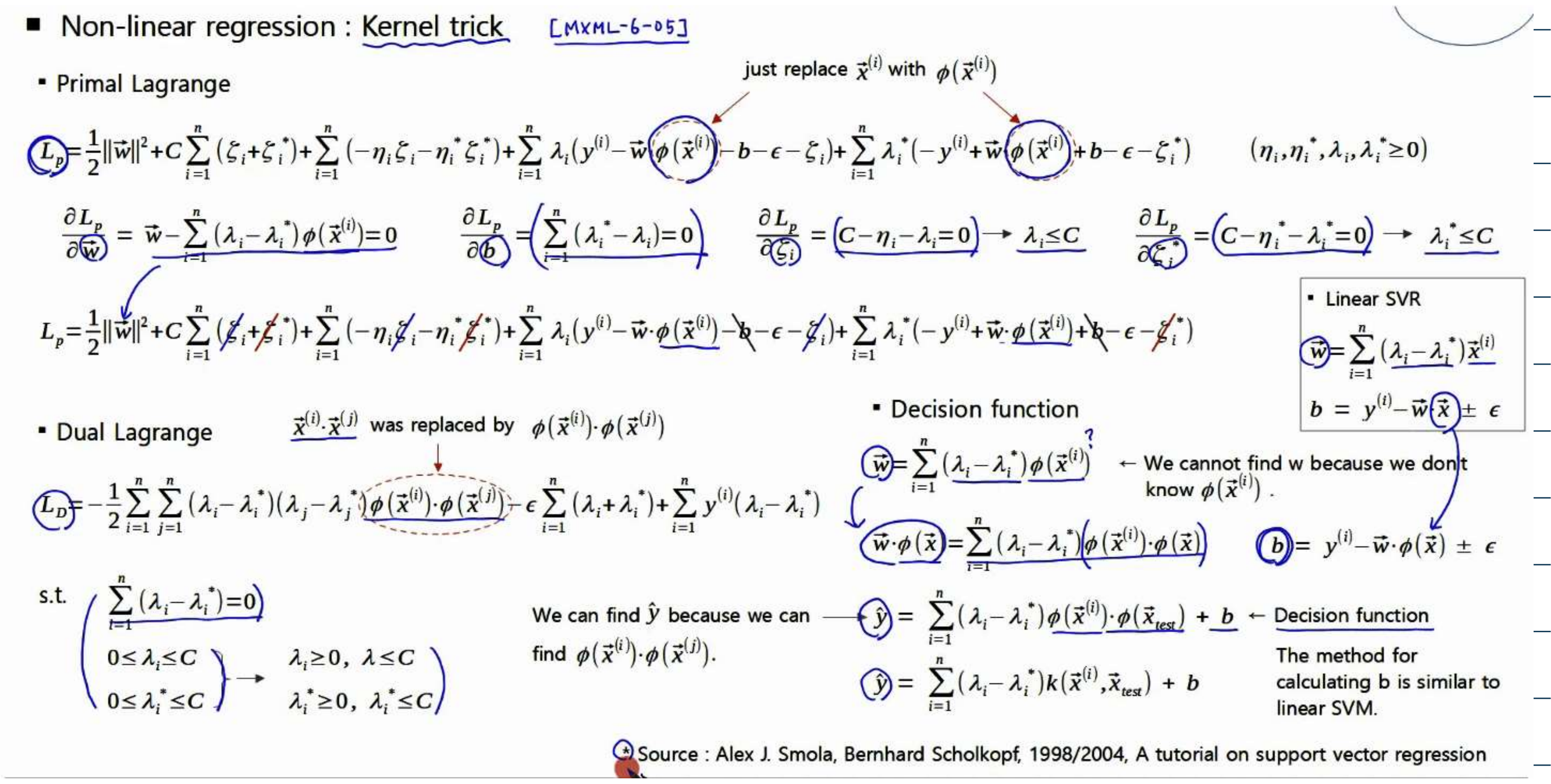
2) $0<\lambda_i^*<C\ \rightarrow\ \zeta_j^*=0\ \rightarrow\ -y^{(i)}+\vec{w}\cdot\vec{x}^{(i)}+b-\epsilon=0$

1) $b=y^{(i)}-\vec{w}\cdot\vec{x}^{(i)}-\epsilon$: For the data points (x, y) that satisfy the condition 0 < λᵢ < C.

2) $b=y^{(i)}-\vec{w}\cdot\vec{x}^{(i)}+\epsilon$: For the data points (x, y) that satisfy the condition 0 < λᵢ < C.

* Take the average of b.

* Source : Alex J. Smola, Bernhard Scholkopf, 1998, A tutorial on support vector regression. 1.4 Computing b, 식 (13)



■ Non-linear regression : Kernel trick [MXML-6-05]

- Primal Lagrange

just replace $\vec{x}^{(i)}$ with $\phi(\vec{x}^{(i)})$

$$L_p=\frac{1}{2}\|\vec{w}\|^2+C\sum_{i=1}^{n}(\zeta_i+\zeta_i^*)+\sum_{i=1}^{n}(-\eta_i\zeta_i-\eta_i^*\zeta_i^*)+\sum_{i=1}^{n}\lambda_i(y^{(i)}-\vec{w}\phi(\vec{x}^{(i)})-b-\epsilon-\zeta_i)+\sum_{i=1}^{n}\lambda_i^*(-y^{(i)}+\vec{w}\phi(\vec{x}^{(i)})+b-\epsilon-\zeta_i^*) \qquad (\eta_i,\eta_i^*,\lambda_i,\lambda_i^*\geq 0)$$

$$\frac{\partial L_p}{\partial \vec{w}} = \vec{w}-\sum_{i=1}^{n}(\lambda_i-\lambda_i^*)\phi(\vec{x}^{(i)})=0 \qquad \frac{\partial L_p}{\partial b}=\sum_{i=1}^{n}(\lambda_i^*-\lambda_i)=0 \qquad \frac{\partial L_p}{\partial \zeta_i}=(C-\eta_i-\lambda_i=0)\rightarrow \lambda_i\leq C \qquad \frac{\partial L_p}{\partial \zeta_i^*}=(C-\eta_i^*-\lambda_i^*=0)\rightarrow \lambda_i^*\leq C$$

$$L_p=\frac{1}{2}\|\vec{w}\|^2+C\sum_{i=1}^{n}(\cancel{\zeta_i}+\cancel{\zeta_i^*})+\sum_{i=1}^{n}(-\eta_i\cancel{\zeta_i}-\eta_i^*\cancel{\zeta_i^*})+\sum_{i=1}^{n}\lambda_i(y^{(i)}-\vec{w}\cdot\phi(\vec{x}^{(i)})-\cancel{b}-\epsilon-\cancel{\zeta_i})+\sum_{i=1}^{n}\lambda_i^*(-y^{(i)}+\vec{w}\cdot\phi(\vec{x}^{(i)})+\cancel{b}-\epsilon-\cancel{\zeta_i^*})$$

- Linear SVR

$$\vec{w}=\sum_{i=1}^{n}(\lambda_i-\lambda_i^*)\vec{x}^{(i)}$$

$$b = y^{(i)}-\vec{w}\vec{x}\pm\epsilon$$

- Dual Lagrange

$\vec{x}^{(i)}\cdot\vec{x}^{(j)}$ was replaced by $\phi(\vec{x}^{(i)})\cdot\phi(\vec{x}^{(j)})$

$$L_D=-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}(\lambda_i-\lambda_i^*)(\lambda_j-\lambda_j^*)\phi(\vec{x}^{(i)})\cdot\phi(\vec{x}^{(j)})-\epsilon\sum_{i=1}^{n}(\lambda_i+\lambda_i^*)+\sum_{i=1}^{n}y^{(i)}(\lambda_i-\lambda_i^*)$$

s.t.
$$\sum_{i=1}^{n}(\lambda_i-\lambda_i^*)=0$$
$$\left.\begin{array}{l}0\leq\lambda_i\leq C\\ 0\leq\lambda_i^*\leq C\end{array}\right\}\rightarrow \begin{array}{l}\lambda_i\geq 0,\ \lambda\leq C\\ \lambda_i^*\geq 0,\ \lambda_i^*\leq C\end{array}$$

- Decision function

$$\vec{w}=\sum_{i=1}^{n}(\lambda_i-\lambda_i^*)\phi(\vec{x}^{(i)})^{?}$$ ← We cannot find w because we don't know $\phi(\vec{x}^{(i)})$.

$$\vec{w}\cdot\phi(\vec{x})=\sum_{i=1}^{n}(\lambda_i-\lambda_i^*)\phi(\vec{x}^{(i)})\cdot\phi(\vec{x}) \qquad b= y^{(i)}-\vec{w}\cdot\phi(\vec{x})\pm\epsilon$$

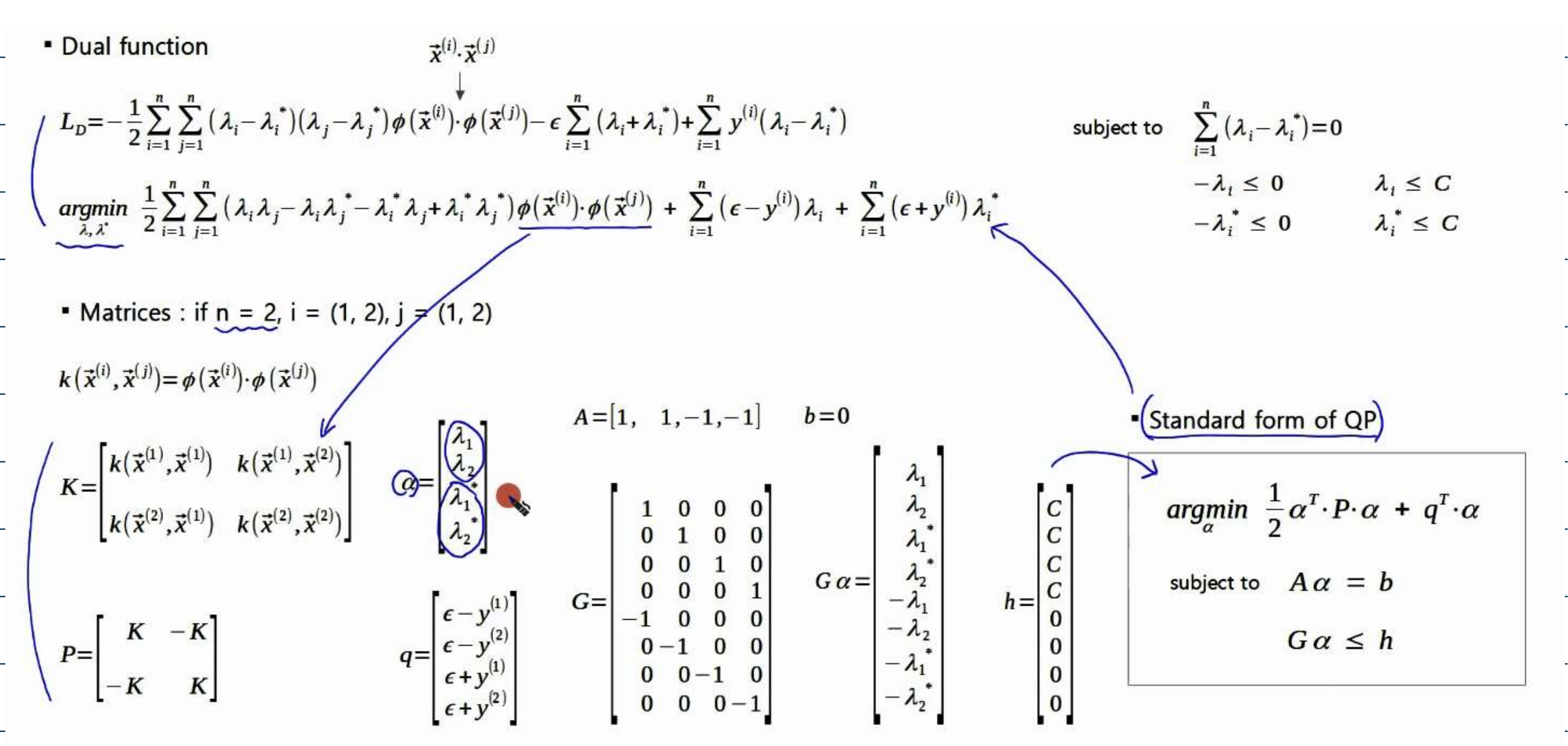
We can find $\hat{y}$ because we can find $\phi(\vec{x}^{(i)})\cdot\phi(\vec{x}^{(j)})$. →

$$\hat{y}=\sum_{i=1}^{n}(\lambda_i-\lambda_i^*)\phi(\vec{x}^{(i)})\cdot\phi(\vec{x}_{test})+b$$ ← Decision function

$$\hat{y}=\sum_{i=1}^{n}(\lambda_i-\lambda_i^*)k(\vec{x}^{(i)},\vec{x}_{test})+b$$

The method for calculating b is similar to linear SVM.

* Source : Alex J. Smola, Bernhard Scholkopf, 1998/2004, A tutorial on support vector regression



- Dual function

$$L_D=-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}(\lambda_i-\lambda_i^*)(\lambda_j-\lambda_j^*)\overset{\vec{x}^{(i)}\cdot\vec{x}^{(j)}\downarrow}{\phi(\vec{x}^{(i)})\cdot\phi(\vec{x}^{(j)})}-\epsilon\sum_{i=1}^{n}(\lambda_i+\lambda_i^*)+\sum_{i=1}^{n}y^{(i)}(\lambda_i-\lambda_i^*)$$

$$\underset{\lambda,\lambda^*}{argmin}\ \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}(\lambda_i\lambda_j-\lambda_i\lambda_j^*-\lambda_i^*\lambda_j+\lambda_i^*\lambda_j^*)\phi(\vec{x}^{(i)})\cdot\phi(\vec{x}^{(j)})+\sum_{i=1}^{n}(\epsilon-y^{(i)})\lambda_i+\sum_{i=1}^{n}(\epsilon+y^{(i)})\lambda_i^*$$

subject to
$$\sum_{i=1}^{n}(\lambda_i-\lambda_i^*)=0$$
$$-\lambda_i\leq 0 \qquad \lambda_i\leq C$$
$$-\lambda_i^*\leq 0 \qquad \lambda_i^*\leq C$$

- Matrices : if n = 2, i = (1, 2), j = (1, 2)

$$k(\vec{x}^{(i)},\vec{x}^{(j)})=\phi(\vec{x}^{(i)})\cdot\phi(\vec{x}^{(j)})$$

$$K=\begin{bmatrix}k(\vec{x}^{(1)},\vec{x}^{(1)}) & k(\vec{x}^{(1)},\vec{x}^{(2)})\\ k(\vec{x}^{(2)},\vec{x}^{(1)}) & k(\vec{x}^{(2)},\vec{x}^{(2)})\end{bmatrix} \qquad \alpha=\begin{bmatrix}\lambda_1\\ \lambda_2\\ \lambda_1^*\\ \lambda_2^*\end{bmatrix}$$

$$A=[1,\ \ 1,-1,-1] \qquad b=0$$

$$P=\begin{bmatrix}K & -K\\ -K & K\end{bmatrix} \qquad q=\begin{bmatrix}\epsilon-y^{(1)}\\ \epsilon-y^{(2)}\\ \epsilon+y^{(1)}\\ \epsilon+y^{(2)}\end{bmatrix} \qquad G=\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\\-1&0&0&0\\0&-1&0&0\\0&0&-1&0\\0&0&0&-1\end{bmatrix} \qquad G\alpha=\begin{bmatrix}\lambda_1\\ \lambda_2\\ \lambda_1^*\\ \lambda_2^*\\ -\lambda_1\\ -\lambda_2\\ -\lambda_1^*\\ -\lambda_2^*\end{bmatrix} \qquad h=\begin{bmatrix}C\\C\\C\\C\\0\\0\\0\\0\end{bmatrix}$$

- Standard form of QP

$$\underset{\alpha}{argmin}\ \frac{1}{2}\alpha^T\cdot P\cdot\alpha+q^T\cdot\alpha$$

subject to $A\alpha = b$

$$G\alpha\leq h$$