CS 3750 Advanced Machine Learning

Support Vector Machines for Regression

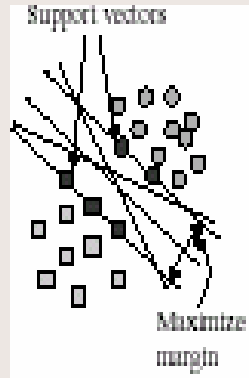
Min Chi
mic31@pitt.edu
Intelligent system program
November 10, 2003

- Introduction
- Basic techniques in SVR
  - Basic linear regression(separable case)
  - Linear ε- Intensive Loss Algorithm(non-separable case)
    - Primal Formulation
    - Dual Formulation
  - Nonlinear regression
    - Kernel Formulation
- Some SVM algorithms
- Conclusion

Support vector machine SVM

•SVM maximize the margin around the separating hyperplane.

•The decision function is fully specified by a subset of the training data, the support vectors.



Introduction

- Transformation with
  - -linear function
  - -nonlinear function-Kernel.
- Nonlinear become linear boundary in the transformed space.
- Aim: To find the optimal Kernel or linear function and corresponding support vectors.

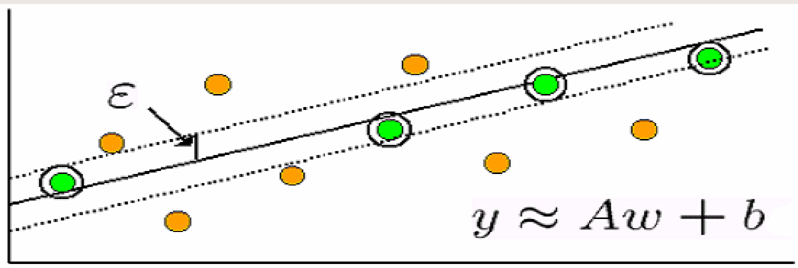
The regression Problem

•Regression=find a function that fits the observations.

• "Close point" may be wrong due to the noise only.

-Line should be influenced by the real data not the noise.

-Ignore the errors from those point which are close.



Duality theory in the convex optimization.

•Uniqueness: Every strictly convex constrained optimization problem has a unique solution.

$$f(x_\lambda) < \lambda f(x_1) + (1-\lambda) f(x_2) \text{ for } x_\lambda = \lambda x_1 + (1-\lambda) x_2$$

•Lagrange Function.

•Dual Objective Function

•Duality Gap

•Karush-Kuhn-Tucker (KKT) conditions.A set of primal and dual variables that is both feasible and satisfies the KKT conditions is the optimal solution.

(i.e. constraint.dual variables=0)

Basic Linear regression (separable case)

- **Training data:**
  $\{(x_1, y_1), \ldots, (x_l, y_l)\}, \quad x \in R^n, y \in R$
- **Our goal is to find a function f(x) that as at most ε deviation from the actually obtained target y_i for all the training data. At the same time as flat as possible.**
- **With a hyperplane(assuming linear model and the data can be separated!)**

$$f(x, a) = \langle w, x \rangle + b$$

Primal regression problem

Linear function f taking the form:

$$f(x) = \langle \omega, x \rangle + b \quad with \quad \omega \in R^n, b \in R \qquad (1)$$

Flatness in the equation (1) means that one seeks small ω. Formally, we can write this problem as a convex optimization problem by:

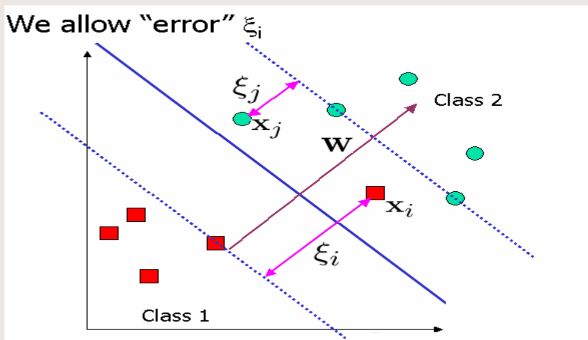
$$\text{minimize} \quad \frac{1}{2}\|w\|^2$$

$$\text{subject to} \quad \begin{cases} y_i - \langle w_i, x_i \rangle - b \le \varepsilon \\ \langle w_i, x_i \rangle + b - y_i \le \varepsilon \end{cases} \qquad (2)$$

All pair (x_i, y_i) with ε precision

Support Vector regression (Vapnik 1995)
(non-separable case)

- Before, we introduce the case that function *f* actually exists that approximates all data pairs with **ε** precision. Sometimes, we may want to allow some errors.



Linear ε -support vector regression

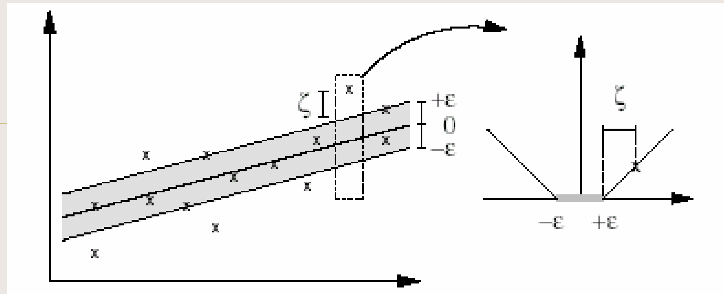
$$\text{minimize} \quad \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}(\xi_i + \xi_i^*)$$

$$\text{subject to} \quad \begin{cases} y_i - \langle w_i, x_i\rangle - b \le \varepsilon + \xi_i \\ \langle w_i, x_i\rangle + b - y_i \le \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \ge 0 \end{cases} \qquad (3)$$

Define $\xi_i = 0$ if there is no error for x_i .

-minimize the error made outside of the tube.

-The parameter C to control the amount of influence of error. C balance the two competing goals.(error and ||w||)



ε-intensive loss function

$$|\xi|_{\varepsilon} = \begin{cases} 0 & \text{for } |\xi| \le \varepsilon \\ |\xi| - \varepsilon & \text{otherwise} \end{cases}$$

The Optimization problem

Lagrangian function will help us to formulate the dual problem

The dual of the problem is :

$$\max\ L := \frac{1}{2}\|\omega\|^2 + C\sum_{i=1}^{l}(\xi_i + \xi_i^*) -$$

$$\sum_{i=1}^{l} a_i(\varepsilon - \xi_i - y_i + \langle \omega, x_i \rangle + b)$$

$$-\sum_{i=1}^{l} a_i^*(\varepsilon + \xi_i^* + y_i - \langle \omega, x_i \rangle - b) - \sum_{i=1}^{l}(\eta_i \xi_i + \eta_i^* \xi_i^*)$$

subject to Lagrange Multiplier : $a_i, a_i^*, \eta_i, \eta_i^* \ge 0$ (5)

primal variable : $\omega, b, \xi_i, \xi_i^*$

It follows from the saddle point condition that the partial derivations of L with respect to the primal variables has to varnish for optimality.

$$\frac{\partial L}{\partial b} = \sum_{i=1}^{l} (a_i^* - a_i) = 0 \qquad (6)$$

$$\frac{\partial L}{\partial \omega} = \omega - \sum_{i=1}^{l} (a_i^* - a_i) x_i = 0 \qquad (7)$$

$$\frac{\partial L}{\partial \xi_i^{(*)}} = C - a_i^{(*)} - \eta_i^{(*)} = 0 \qquad (8)$$

calculation

$$L = \frac{1}{2}\langle w, w\rangle + \sum_{i=1}^{l} C\xi_i + \sum_{i=1}^{l} C\xi_i^*$$

$$-\sum_{i=1}^{l} a_i\varepsilon - \sum_{i=1}^{l} a_i\xi_i - \sum_{i=1}^{l} a_i y_i - \sum_{i=1}^{l} a_i\langle \omega, x_i\rangle + \sum_{i=1}^{l} a_i b$$

$$-\sum_{i=1}^{l} a_i^*\varepsilon - \sum_{i=1}^{l} a_i^*\xi_i^* - \sum_{i=1}^{l} a_i^* y_i + \sum_{i=1}^{l} a_i^*\langle \omega, x_i\rangle + \sum_{i=1}^{l} a_i^* b$$

$$-\sum_{i=1}^{l} \eta_i\xi_i - \sum_{i=1}^{l} \eta_i^*\xi_i^*$$

$$L=\frac{1}{2}\langle w,w\rangle+\sum_{i=1}^{l}\xi_i\underbrace{(C-\eta_i-a_i)}_{=0(\text{from (8)},\ C-\eta_i^{(*)}-a_i^{(*)}=0)}+$$

$$\sum_{i=1}^{l}\xi_i^*\underbrace{(C-\eta_i^*-a_i^*)}_{=0(\text{from (8)},\ C-\eta_i^{(*)}-a_i^{(*)}=0)}-\sum_{i=1}^{l}(a_i+a_i^*)\varepsilon-\sum_{i=1}^{l}(a_i+a_i^*)y_i$$

$$-\sum_{i=1}^{l}\underbrace{(a_i-a_i^*)\langle\omega,x_i\rangle}_{=\langle w,w\rangle(\text{From (7)},\ \omega=\sum_{i=1}^{l}(a_i+a_i^*)x_i)}+\sum_{i=1}^{l}\underbrace{(a_i^*-a_i)b}_{=0(\text{From (6)},\ \sum_{i=1}^{l}(a_i^*-a_i)=0)}$$

$$L=-\frac{1}{2}\langle w,w\rangle-\sum_{i=1}^{l}(a_i+a_i^*)\varepsilon-\sum_{i=1}^{l}(a_i+a_i^*)y_i$$

$$=-\frac{1}{2}\sum_{i=1}^{l}(a_i-a_i^*)(a_j-a_j^*)\langle x_i,x_j\rangle-$$

$$\sum_{i=1}^{l}(a_i+a_i^*)\varepsilon-\sum_{i=1}^{l}(a_i+a_i^*)y_i$$

$$\text{subject to}\quad:\begin{cases}\sum_{i=1}^{l}(a_i-a_i^*)=0\\ a_i,a_i^*\in[0,C]\end{cases}\qquad(9)$$

Results.

From the equation (7)

$$\omega = \sum_{i=1}^{l} (a_i - a_i^*) x_i$$

We can get:

$$f(x) = \sum_{i=1}^{l} (a_i - a_i^*) \langle x_i, x \rangle + b \qquad (10)$$

How to computing b?

- At the KKT(Karush-Kuhn-Tucker) conditions: at the optimal solution the product between dual variables and constraints has to vanish. This means that the Lagrange multipliers will only be non-zero for points outside the ε band. Thus these points are the support vectors.In the SV case:

$$a_i(\varepsilon - \xi_i - y_i + \langle \omega, x_i \rangle + b) = 0$$

$$a_i^*(\varepsilon + \xi_i^* + y_i - \langle \omega, x_i \rangle - b) = 0 \qquad (11)$$

$$(C - a_i)\xi_i = 0$$

$$(C - a_i^*)\xi_i^* = 0 \qquad (12)$$

How to computing b?

Firstly, only samples (x_i, y_i) with corresponding $a_i^{(*)} = C$ lie outside the the ε-insensitive tube around *f*.

Secondly, $a_i a_i^* = 0$ there can never be a set of dual variables a_i, a_i^* which are both simultaneously nonzero.

Finally, for $a_i^{(*)} \in (0, C)$ we have $\xi_i^{(*)} = 0$.

Hence b can be computed as follows :

$$b = y_i - \langle \omega, x_i \rangle - \varepsilon \quad \text{for } a_i \in (0, C)$$

$$b = y_i - \langle \omega, x_i \rangle + \varepsilon \quad \text{for } a_i^* \in (0, C) \qquad (13)$$

Nonlinear SVR

•Key idea:transform xi to a higher dimension to make life easier.

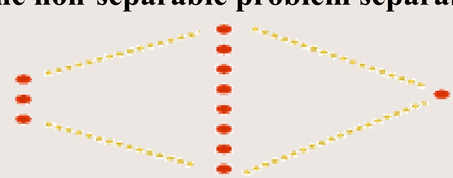
-input space: the space xi are in.

-feature space: the space of Φ(xi) after transformation.

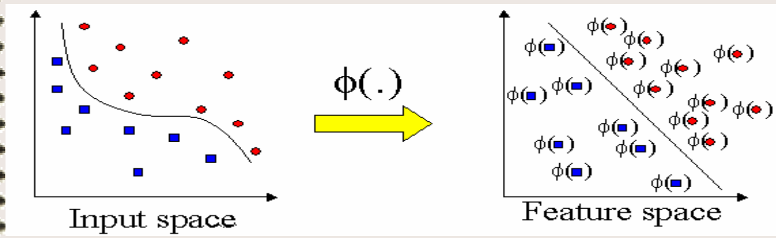
•Why transform?

- Linear operation in the feature space is equivalent to the nonlinear operation in the input space.

-make the non-separable problem separable.



(continue)



Possible problems of the transformation:

- High computation burden

SVM solved the problem:

- Kernel tricks for efficient computation.

Example Transformation

- Define the Kernel Function:

$$k\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}\right) = (1 + x_1 y_1 + x_2 y_2)^2$$

- Consider the following transformation:

$$\Phi\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right) = (1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, x_2^2, \sqrt{2}x_1 x_2)$$

$$\Phi\left(\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}\right) = (1, \sqrt{2}y_1, \sqrt{2}y_2, y_1^2, y_2^2, \sqrt{2}y_1 y_2)$$

$$\left\langle \Phi\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right), \Phi\left(\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}\right)\right\rangle = (1 + x_1 y_1 + x_2 y_2)^2$$

$$= k\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}\right)$$

So the inner product can be computed by k without going through the mapping $\Phi(.)$.

Kernel Trick

The relationship between the Kernel funtion k and the mapping $\Phi(.)$ is:

$k(x,y) = \langle \Phi(x), \Phi(y) \rangle$.

- This is known as the Kernel trick.

•We choose *k* instead of choosing Φ(.)

•*K(x,y)* need to satisfy a technical condition (Mercer Conditions) in order for Φ(.) to exist.

Kernel

Which functions $k(x, x')$ correspond to a dot product in the feature space?

$$K(x_i, x_j) = \Phi(x_i).\Phi(x_j)$$

Hilbert Schmidt Theory:

$K(x_i, x_j)$ **is a symmetric function.**

Mercer's conditions:

$$K(x_i, x_j) = \sum_{m=1}^{\infty} a_m \Phi(x_i).\Phi(x_j) \text{ if}$$

$$\iint K(x, x') f(x) f(x') dxdx' \geq 0,$$

$$\int f^2(x) dx < \infty$$

Linear combination of Kernels:

$$k(x,x') := c_1 k_1(x,x') + c_2 k_2(x,x')$$

Integral of Kernels:

$$k(x,x') := \int s(x,z)s(x',z)\,dz$$

Smola, Scholkopf and Muller(1998):

$$k(x,x') := k(x - x') \quad \text{if}$$

$$F[k](\omega) = (2\pi)^{-\frac{d}{2}} \int e^{-i\langle \omega, x\rangle} k(x)\,dx \geq 0$$

Burges(1999): $k(x,x') := k(\langle x,x'\rangle)$

$$k(\xi) \geq 0,$$

$$\partial_\xi k(\xi) \geq 0,$$

$$\partial_\xi k(\xi) + \xi \partial_\xi^2 k(\xi) \geq 0$$

Non-linear SVR algorithm

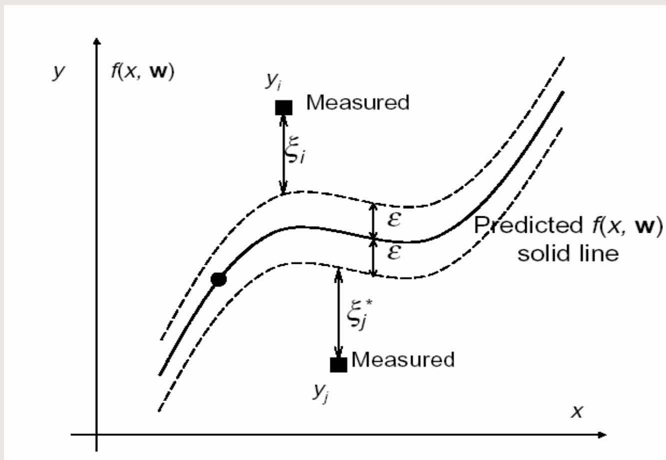
Primal problem:

minimize $\quad \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}(\xi_i + \xi_i^*)$

subject to $\quad \begin{cases} \mathrm{y_i} - \langle w_i, \Phi(x_i)\rangle - b \leq \varepsilon + \xi_i \\ \langle w_i, \Phi(x_i)\rangle + b - \mathrm{y_i} \leq \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0 \end{cases} \qquad (3)$

$$\omega = \sum_{i=1}^{l}(a_i - a_i^*)\Phi(x_i)$$

Parameters used in SV Regression



Non-linear SVR algorithm

similar with (9)

$$\text{maximize}\begin{cases} L = -\frac{1}{2}\sum_{i=1}^{l}(a_i - a_i^*)(a_j - a_j^*)k(x_i, x_j) - \\ \sum_{i=1}^{l}(a_i + a_i^*)\varepsilon - \sum_{i=1}^{l}(a_i + a_i^*)y_i \end{cases} \quad (14)$$

$$\text{subject to:}\begin{cases} \sum_{i=1}^{l}(a_i - a_i^*) = 0 \\ a_i, a_i^* \in [0, C] \end{cases}$$

If $k(x, x') := \langle \Phi(x), \Phi(x') \rangle$. Similar with (10):

$$\omega = \sum_{i=1}^{l}(a_i - a_i^*)\Phi(x_i) \quad \text{and } f(x) = \sum_{i=1}^{l}(a_i - a_i^*)k(x_i, x) + b \quad (15)$$

Cost Function

Training data:

$$\{(x_1, y_1), \ldots, (x_l, y_l)\}, \quad x \in R^n, \; y \in R$$

We assume that the training data has been drawn iid from some probability distribution p(*x,y*). Our goal is to find a function f(x) that minimizes a risk functional.

$$R[f] = \int c(x, y, f(x))\, dp(x, y)$$

A possible approximation:

$$R_{emp}[f] := \frac{1}{l} \sum_{i=1}^{l} c(x_i, y_i, f(x_i))$$

Add a capacity control term:

$$R_{reg}[f] := R_{emp}[f] + \frac{\lambda}{2} \|\omega\|^2$$

$\lambda > 0$ called regulation constant.

One hand, we want to avoid using a very complicated function c as this may lead to difficult optimization; on the other hand, we should use the cost function that suits data best. Under the assumption that the data were idd and generated by function plus addictive noise.

$$c(x, y, f(x)) = -\log p(y - f(x))$$

	loss function	density model
ε-insensitive	$c(\xi) = \|\xi\|_\varepsilon$	$p(\xi) = \frac{1}{2(1+\varepsilon)} \exp(-\|\xi\|_\varepsilon)$
Laplacian	$c(\xi) = \|\xi\|$	$p(\xi) = \frac{1}{2} \exp(-\|\xi\|)$
Gaussian	$c(\xi) = \frac{1}{2}\xi^2$	$p(\xi) = \frac{1}{\sqrt{2\pi}} \exp(-\frac{\xi^2}{2})$
Huber's robust loss	$c(\xi) = \begin{cases} \frac{1}{2\sigma}(\xi)^2 & \text{if } \|\xi\| \le \sigma \\ \|\xi\| - \frac{\sigma}{2} & \text{otherwise} \end{cases}$	$p(\xi) \propto \begin{cases} \exp(-\frac{\xi^2}{2\sigma}) & \text{if } \|\xi\| \le \sigma \\ \exp(\frac{\sigma}{2} - \|\xi\|) & \text{otherwise} \end{cases}$
Polynomial	$c(\xi) = \frac{1}{p}\|\xi\|^p$	$p(\xi) = \frac{p}{2\Gamma(1/p)} \exp(-\|\xi\|^p)$
Piecewise polynomial	$c(\xi) = \begin{cases} \frac{1}{p\sigma^{p-1}}(\xi)^p & \text{if } \|\xi\| \le \sigma \\ \|\xi\| - \sigma\frac{p-1}{p} & \text{otherwise} \end{cases}$	$p(\xi) \propto \begin{cases} \exp(-\frac{\xi^p}{p\sigma^{p-1}}) & \text{if } \|\xi\| \le \sigma \\ \exp(\sigma\frac{p-1}{p} - \|\xi\|) & \text{otherwise} \end{cases}$

Table 1 Common loss functions and corresponding density models

For the loss function:

$$c(x, y, f(x)) = \begin{cases} 0 & \text{for } |y - f(x)| \le \varepsilon \\ \tilde{c}(|y - f(x)| - \varepsilon) & \text{otherwise} \end{cases} \qquad (15)$$

Compare with $\varepsilon -$intensive loss function

$$|\xi|_\varepsilon = \begin{cases} 0 & \text{for } |\xi| \le \varepsilon \\ |\xi| - \varepsilon & \text{otherwise} \end{cases}$$

•Extend the special choice to more general convex cost functions.

• Moreover, we might choose different cost functions $\tilde{c}_i, \tilde{c}_i^*$ and different values of $\varepsilon, \varepsilon^*$ for each sample. Similar with (3), we can get:

By standard Lagrange multiplier technique, exactly the same manner as in the ε-intensive loss function.

$$\text{minimize} \quad \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}(\tilde{c}(\xi_i) + \tilde{c}(\xi_i^*))$$

$$\text{subject to} \quad \begin{cases} \mathrm{y_i} - \langle w_i, x_i \rangle - b \le \varepsilon + \xi_i \\ \langle w_i, x_i \rangle + b - \mathrm{y_i} \le \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \ge 0 \end{cases} \qquad (16)$$

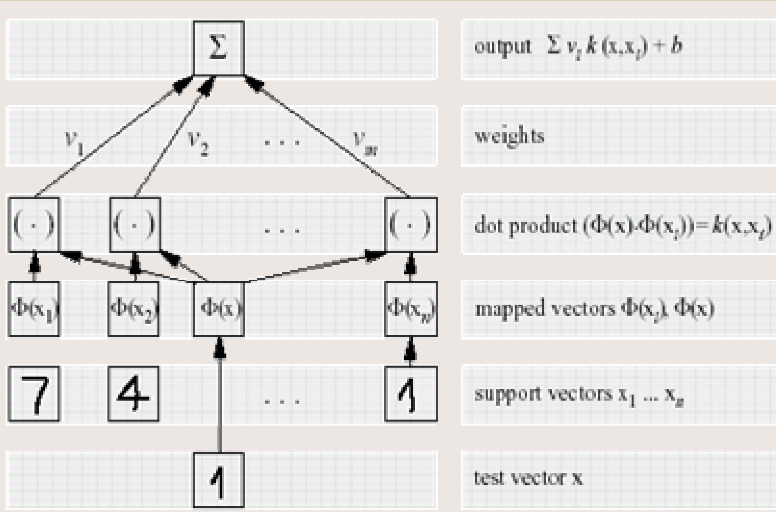
$$L = -\frac{1}{2}\sum_{i=1}^{l}(a_i - a_i^*)(a_j - a_j^*)\langle x_i, x_j\rangle +$$

$$\sum_{i=1}^{l}\left(y_i(a_i - a_i^*) - \varepsilon(a_i + a_i^*) + C(\mathrm{T}(\xi_i) + \mathrm{T}(\xi_i^*))\right)$$

$$\text{where}\quad \begin{cases} \omega = \sum_{i=1}^{l}(a_i - a_i^*)x_i \\ \mathrm{T}(\xi) := \tilde{c}(\xi) - \xi\partial_{\xi}\tilde{c}(\xi) \end{cases} \qquad (17)$$

$$\text{subject to}\quad : \begin{cases} \sum_{i=1}^{l}(a_i - a_i^*) = 0 \\ a \in [0, C\partial_{\xi}\tilde{c}(\xi)] \\ \xi = \inf\{\xi \mid C\partial_{\xi}\tilde{c}(\xi) \geq a\} \\ \xi \geq 0 \end{cases} \qquad (18)$$

Architecture



Algorithms- Interior Point Algorithms.

variable a denote vector and α_i denotes the i – th component.

minimize $\frac{1}{2}q(\alpha) + c.a$

subject to $A\alpha = b \quad l \le \alpha \le \mu \quad c, a, l, \mu \in R^n, A \in R, b \in R^m$

- Add slack variables:

$$\text{minimize} \quad \frac{1}{2}q(\alpha) + c.a$$

$$\text{subject to } A\alpha = b \; a\text{-}g = l, \; a + t = \mu \quad g, t \ge 0$$

- Write the Wolfe dual.
- Get the KKT conditions
- Solve the Equations

SMO(Sequential minimal optimization)

-each iteration,SMO chooses only two α_i ,and find optimal value, updates

the SVM to reflect new optimal value

- 3 components to SMO

1. **Analytic method to solve for two lagrange multiplier**
2. **Heuristic for choosing**
3. **A method for computing bias b**

Other Algorithms

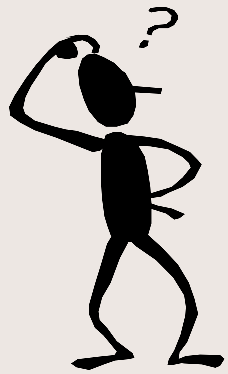
- **Subset selection algorithms**
- **Inverse problems.(suitable for specific settings)**
- **Convex combination and l_1-norms.(different ways of measuring capacity and reduction to the linear programming)**
- **Semiparametric modeling.(different ways of controlling capacity and different classes).**

conclusion

- **Linear operation in the feature space is equivalent to the nonlinear operation in the input space.**
- **key concepts of SVM**
  - **optimization**
  - **kernel trick**
- **There are still a lot of open issues in SVR.**

  Both time complexity & storage capacity problem are
  - **increasing as train data increase**
  - **The choice of kernel function : there are no guidelines**

Question?



Thank you!