Support Vector Machines (SVM)

- optimal separating hyperplane

. linearly separable case for binary classification

l samples of training data:

$$(x_1, y_1), (x_2, y_2), \cdots, (x_l, y_l), \quad x \in R^n, \quad y \in \{-1,+1\}$$

. hyperplane decision function:

$$D(x) = (w \cdot x) + w_0$$

$$y_i[(w \cdot x_i) + w_0] \geqq 1, \quad i = 1, \cdots, l$$

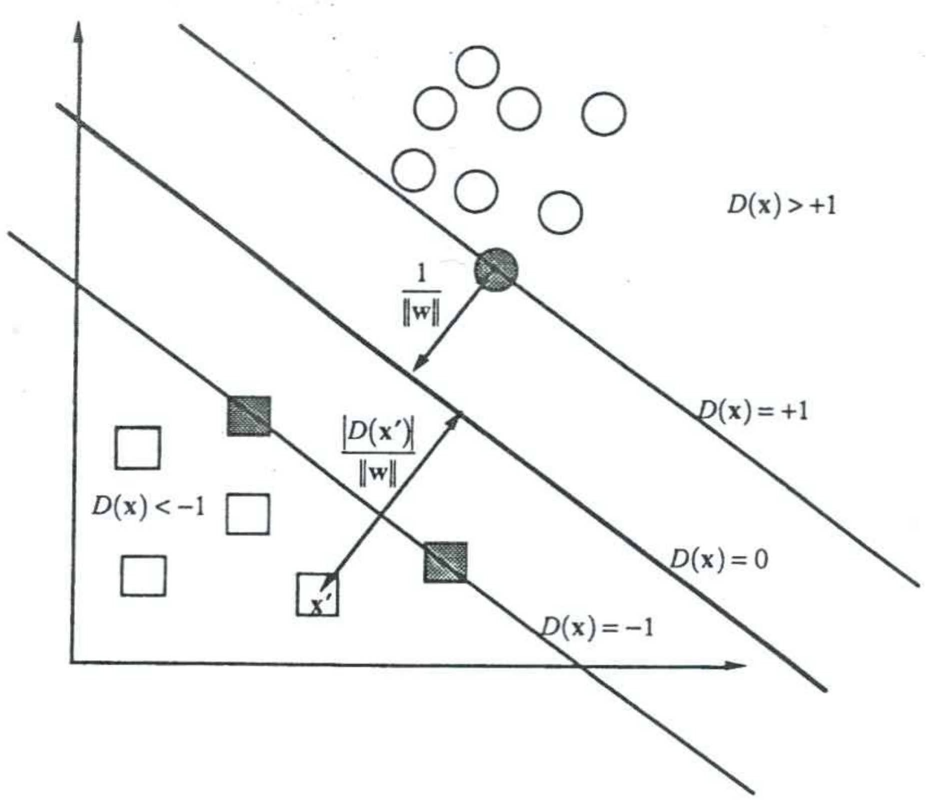
. margin ρ: the minimal distance from the separating hyperplnae (s. h.) to the closest data

. optimal s. h.: the s. h. in which the margin ρ is maximum.

. distance between s. h. and a sample x': $\dfrac{|D(x')|}{\| w \|}$

. all samples obey $\dfrac{y_k D(x_k)}{\| w \|} \geqq \rho, \quad k = 1, \cdots, l.$

. support vector (s.v.): the sample that exists at the margin



- VC dimension of Perceptrons

Theorem (Vapnik, 1998):

- Let $x^* = (x_1, \cdots, x_l)$ be a set of l vectors in R^n.
- For any hyperplane $(x \cdot w) + w_0 = 0$ in R^n, consider the corresponding cannonical hyperplane defined by the set X^* such that $\infty_{x \in X^*} |(x \cdot w) + w_0| = 1$.
- A subset of cannonical hyperplane defined on $X^* \subset R^n$ such that $|x| \leqq D$, $x \in X^*$ satisfying the constraint $|w| \leqq A$ has the VCD h bounded as follows:

$$h \leqq \min([D^2 A^2], n) + 1 \quad \text{or} \quad h \leqq \min(\left[\frac{D^2}{\rho^2}\right], n) + 1.$$

Theorem (Vapnik, 1998):

With the probability at least $1-\delta$, one can assert that

$$R(\alpha_l) \leqq \frac{m}{l} + \frac{\epsilon}{2}\left(1 + \sqrt{\frac{4m}{\epsilon}}\right)$$

where

$$\epsilon = 4\frac{h\left(1+\ln\frac{2l}{h}\right) - \ln\frac{\delta}{4}}{l},$$

$m =$ the number of training samples that are not separated correctly, and

$h =$ the upper bound of the VCD.

- support vector machine (SVM) learning

. Learning problems is changed to the quadratic optimization problems:

Determine w and w_0 that minimizes the functional $\eta(w)$, that is,

$$\min_w \eta(w) = \frac{1}{2} \| w \|^2$$

subject to

$$y_i[(w \cdot x_i) + w_0] \geqq 1 \quad \text{for } i = 1, \cdots, l$$

. Dual problem:

- If the cost and constraint functions are strictly convex, solving the dual problem is equivalent to solving the original problem.

- Functions are convex if
$$f(\alpha x_1 + (1-\alpha)x_2) \leqq \alpha f(x_1) + (1-\alpha)f(x_2) \quad \forall x_1, x_2 \in C,\ 0 < \alpha < 1$$
example: quadratic functions

- Procedure of formulating the dual problem

(1) Constructing the Lagrangian function:

$$Q(w, w_0, \alpha) = \frac{1}{2}(w \cdot w) - \sum_{i=1}^{l} \alpha_i \{ y_i [(w \cdot x) + w_0] - 1 \}$$

where α_i is Lagrangian multiplier.

(2) Searching for the optimal conditions:

(a) $\dfrac{\partial Q(w^*, w_0^*, \alpha^*)}{\partial w} = 0$

-> $w^* = \sum_{i=1}^{l} \alpha_i^* y_i x_i$, $\alpha_i^* \geqq 0$, for $i = 1, \cdots, l$.

(b) $\dfrac{\partial Q(w^*, w_0^*, \alpha^*)}{\partial w_0} = 0$

-> $\sum_{i=1}^{l} \alpha_i^* y_i^* = 0$, $\alpha_i^* \geqq 0$, for $i = 1, \cdots, l$.

(3) Formulating the dual problem:

Find the parameters α_i for $i=1, \cdots, l$ maximizing the functional

$$Q(\alpha)=\sum_{i=1}^{l}\alpha_i-\frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j y_i y_j (x_i \cdot x_j),\ \ \alpha_i^* \geqq 0,\ \text{for } i=1, \cdots, l.$$

subject to

$$\sum_{i=1}^{l}\alpha_i y_i=0,\ \ \alpha_i \geqq 0,\ \text{for } i=1, \cdots, l.$$

- Kuhn-Tucker Theorem:

Any parameter α_i^* is non-zero only if

$y_i[(w \cdot x_i)+w_0]=1$, for $i=1, \cdots, l$, that is,

$\alpha_i^*\{y_i[(w^* \cdot x_i)+w_0^*]-1\}=0$, for $i=1, \cdots, l$.

-> the data corresponding to non-zero α_i^* are support vectors.

-> the resulting equation for s. h.:

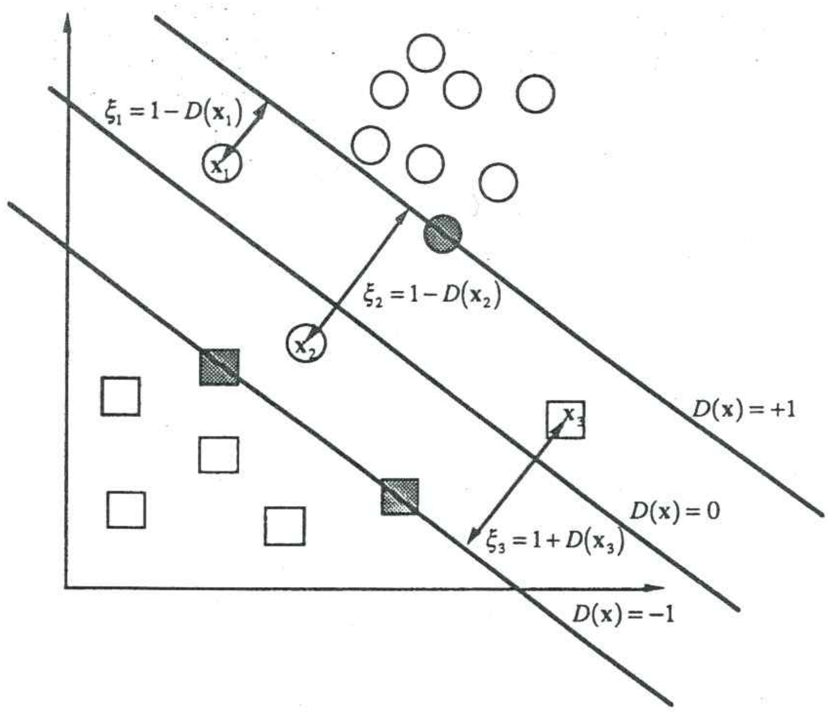
$$D(x)=\sum_{i=1}^{l}\alpha_i^* y_i (x \cdot x_i)+w_0^*$$

- non-separable problems

. For non-separable problems,

apply the positive slack variables ξ_i, that is,

$y_i[(w \cdot x_i)+w_0] \geqq 1-\xi_i$, for $i=1, \cdots, l$



For a training sample x_i, the slack variable ξ_i is the deviation from the margin border corresponding to the class $y_i (= D(x_i))$.

if $\xi_i > 0$, non-separable sample

if $\xi_i > 1$, misclassified sample

. optimization problem with slack variables:

$$\min_w \frac{c}{l} \sum_{i=1}^{l} \xi_i + \frac{1}{2} \parallel w \parallel^2$$

where c is a positive constant.

subject to

$$y_i [(w \cdot x_i) + w_0] \geqq 1 - \xi_i \quad \text{for } i = 1, \cdots, l.$$

Applying the dual problem procedure, we get the following dual problem:

$$\max_{\alpha} Q(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j)$$

subject to

$$\sum_{i=1}^{l} \alpha_i y_i = 0 \text{ and } 0 \leqq \alpha_i \leqq \frac{c}{l} \quad \text{for } i = 1, \cdots, l.$$

- kernel basis functions

. constructing the nonlinear s. h.

. decision function in linear case:

$$D(x) = \sum_{i=1}^{l} \alpha_i^* y_i (x \cdot x_i) + w_0^*$$

Here, $(x \cdot x_i)$ is replaced by a kernel function $K(x, x_i)$.

. condition for kernel functions (Mercer's theorem)

A kernel is a continuous function that maps

$$K: [a,b] \times [a,b] \rightarrow R$$

such that $K(x,s) = K(s,x)$.

K is said to be non-negative definite if and only if

$$\sum_{i=1}^{n}\sum_{j=1}^{n}K(x_i,x_j)c_ic_j \geqq 0$$

for all finite sequences of points $x_1, \cdots, x_n$ of $[a,b]$ and all choices of real numbers $c_1, \cdots, c_n$.

Associated to K is a linear operator on functions defined by the integral

$$[T_K\phi](x) = \int_a^b K(x,s)\phi(s)ds$$

We assume that ϕ can range through the space $L^2[a,b]$ of square integrable real-valued functions. Since T is a linear operator, we can talk about eigenvalues and eigenfunctions of T.

Mercer's theorem:

Suppose K is a continuous symmetric non-negative definite kernel. Then, there is an orthonormal basis $\{e_i\}$ of $L^2[a,b]$ consisting of eigenfunctions of T_K such that corresponding sequence of eignevalues $\{\lambda_i\}$ is non-negative.

The eigenfunctions corresponding to non-zero eigenvalues are continuous on $[a,b]$ and K has the representation

$$K(s,t) = \sum_{j=1}^{\infty}\lambda_j e_j(s)e_j(t)$$

where the convergence is absolute and uniform.

Examples of kernel functions:

(a) polynomials of degree p: $K(x,x') = [(x \cdot x')+1]^p$

(b) radial basis functions: $K(x,x') = \exp\left(-\frac{|x-x'|^2}{\sigma^2}\right)$

(c) sigmoid functions: $K(x,x') = \tanh(\nu(x \cdot x')+a)$

Dual problem:

$$\max_\alpha Q(\alpha) = \sum_{i=1}^{l} \alpha - \frac{1}{2}\sum_i \sum_j \alpha_i \alpha_j y_i y_j K(x_i,x_j)$$

subject to

$$\sum_{i=1}^{l} \alpha_i y_i = 0 \quad \text{and} \quad 0 \leqq \alpha \leqq \frac{c}{l} \quad \text{for } i=1,\cdots,l$$

The resulting equation for s. h.

$$D(x) = \sum_{i=1}^{l} \alpha_i^* y_i K(x,x_i).$$

- SVMs for regression

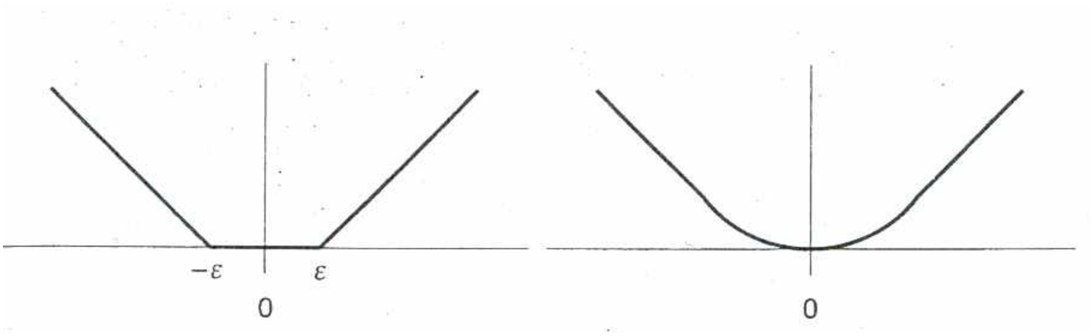
. Estimation function:

$$f(x,w) = \sum_{i=1}^{m} w_i K(x,x_i)$$

where $K(x,x_i)$ represents the kernel function located at x_i.

. Vapnik's e-sensitive loss function:

$$L_e(y,f(x,w)) = \begin{cases} 0 & \text{if } |y-f(x,w)| \leqq e \\ |y-f(x,w)|-e & otherwise \end{cases}$$



. learning problem:

finding w that minimizes

$R_{emp}(w) = \frac{1}{l}\sum_{i=1}^{l} L_e(y,f(x,w))$ under the constraint $\| w \|^2 \leqq C$

. quadratic problem:

$$\min_w \frac{c}{l}\left(\sum_{i=1}^{l}\xi_i + \sum_{i=1}^{l}\xi_i'\right) + \frac{1}{2}\| w \|^2$$

subject to

$$y_i - \sum_{i=1}^{l} w_i K(x,x_i) \leqq e+\xi_i', \quad \sum_{i=1}^{l} w_i K(x,x_i) - y_i \leqq e+\xi_i,$$

$\xi_i \geqq 0$ and $\xi_i' \geqq 0$

. dual problem:

$$\max_{\alpha,\beta} Q(\alpha,\beta) = -e\sum_{i=1}^{l}(\alpha_i+\beta_i)+\sum_{i=1}^{l}y_i(\alpha_i-\beta_i)$$
$$-\frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}(\alpha_i-\beta_i)(\alpha_j-\beta_j)K(x_i,x_j)$$

subject to

$$\sum_{i=1}^{l}\alpha_i=\sum_{i=1}^{l}\beta_i,\ 0\leqq\alpha_i\leqq\frac{c}{l}$$ and $0\leqq\beta_i\leqq\frac{c}{l}$ for $i=1,\cdots,l$.

. the final estimation function

$$f(x)=\sum_{i=1}^{l}(\alpha_i^*-\beta_i^*)K(x,x_i)$$

. the generalization bound of SVM using the non-negative loss function:

Let

for $p>2$.

Then, with the probability at least $1-\delta$

$$R(\alpha_l)\leqq e+\frac{R_{emp}(\alpha_l)-e}{(1+a(p)\tau\sqrt{\epsilon})_+}$$

where

$$a(p) = \sqrt[p]{\frac{1}{2}\left(\frac{p-1}{p-1}\right)^{p-1}}, \quad \epsilon = 4\frac{h_n(1+\ln\frac{2l}{h_n}) - \ln\frac{\delta}{4}}{l}, \text{ and}$$

h_n is the VCD of

$$S_n = \left\{L_e(y, f(x,w)) \mid \| w \|^2 \leq C\right\}.$$

- multi-class SVMs

. k-class pattern recognition

Constructing a decision function given l i.i.d. samples:

$$(x_1, y_1), \cdots, (x_l, y_l)$$

where x_i, $i = 1, \cdots, l$ are vectors of length d and

$y_i \in \{1, \cdots, k\}$ are classes of samples.

Here, the loss function is given by

$$L(y, f(x,w)) = \begin{cases} 0 & \text{if } \; y = f(x,w) \\ 1 & otherwise \end{cases}$$

where w is a parameter vector.

. Example: binary classification

$k=2, \quad y_i \in \{-1,+1\}$.

(1) optimization problem:

$$\min_w \phi(w,\xi) = \frac{1}{2}(w \cdot w) + c\sum_{i=1}^{l}\xi_i$$

subject to

$y_i((w \cdot x_i)+b) \geqq 1-\xi_i$ for $i=1, \cdots, l$ and

$\xi_i \geqq 0$ for $i=1, \cdots, l$.

(2) dual problem:

$$\max_\alpha Q(\alpha) = \sum_{i=1}^{l}\alpha_i - \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l} y_i y_j \alpha_i \alpha_j (x_i \cdot x_j)$$

subject to

$0 \leqq \alpha_i \leqq c$ for $i=1, \cdots, l$ and $\sum_{i=1}^{l}\alpha_i y_i = 0$.

(3) the optimal decision function:

$$f(x) = sign\left[\sum_{i=1}^{l}\alpha_i^* y_i (x \cdot x_i) + b^*\right].$$

. one-against-the rest method

The problem is converted into k binary classification problems.
For the ith class

$y_j = 1$ if x_j belongs to the ith class; -1 otherwise.

That is, we have k l-variable quadratic optimization problems.

In general, this method gives good performance but it is computationally expensive and SVMs have many overlapped support vectors.

. one-against-one method

This method selects binary classifier among k classes, that is, we have ${}_kC_2 = k(k-1)/2$ classifiers.

On the average, each class has l/k samples. This implies that this method needs to solve $(k(k-1)/2)$ $(2l/k)$ variable quadratic optimization problem.

For each classifier, small number of samples is need to be trained compared to the one-against-one method.
Overall computational complexity is same as the one-against-one method. However, if we use systematic reduction of samples such as tree structure, further reduction of computational complexity is possible.

. k-class SVMs

General case of the binary class SVMs.

(1) optimization problem:

$$\min_w \phi(w,\xi) = \frac{1}{2}\sum_{m=1}^{k}(w_m \cdot w_m) + c\sum_{i=1}^{l}\sum_{m \neq y_i}^{k}\xi_i^m$$

subject to

$(w_{y_i} \cdot x_i) + b_{y_i} \geqq (w_m \cdot x_i) + b_m + 2 - \xi_i^m$ for $i = 1, \cdots, l$ and

$\xi_i^m \geqq 0$ for $i = 1, \cdots, l.$

That is, we need to solve 1 kl variable quadratic optimization problem.

(2) dual problem:

$$\max_\alpha Q(\alpha) = 2\sum_{i,m \neq y_i}\alpha_i^m + \sum_{i,j,m \neq y_i}\left[-\frac{1}{2}c_j^{y_i}A_iA_j + \alpha_i^m\alpha_j^{y_i} - \frac{1}{2}\alpha_i^m\alpha_j^m\right](x_i \cdot x_j)$$

subject to

$\sum_{i=1}^{l}\alpha_i^n = \sum_{i=1}^{l}c_i^nA_i$ for $n = 1, \cdots, k,$

$0 \leqq \alpha_i^m \leqq c,$ and $\alpha_i^{y_i} = 0$ for $i = 1, \cdots, l$

(3) the optimal decision function:

$$D(x) = \arg\max_n\left[\sum_{i=1}^{l}(c_i^nA_i - \alpha_i^n)(x_i \cdot x) + b_n\right]$$

- reducing the computational complexity in SVM learning

. dual problem:

$$\max_{\alpha} Q(\alpha) = \alpha^T \cdot 1 - \frac{1}{2}\alpha^T D\alpha$$

subject to

$\alpha^T \cdot y = 0$ and $0 \leqq \alpha \leqq c$

where

$\alpha = [\alpha_1, \cdots, \alpha_l]^T$, $D = [d_{ij}]$, and

$d_{ij} = y_i y_j K(x_i, x_j)$.

If $l = 10K$ samples, we need l^2 memory for writing D.

each sample takes 4 bytes -> 1.6 Gbytes to store D.

. chunking method:

reducing the size of samples

algorithm:

Given training set S

Select an arbitrary working set (chunk) $\hat{S} \subset S$.

Repeat

solve the optimization problem on $\hat{S}$.

select a new working set (chunk) from data not satisfying Kuhn-Tucker conditions.

until stopping criterion satisfied.

Return α.

. decomposition method

reducing the size of α: divide α into two sets,

a working set α_W and the remaining set α_R, that is,

$\alpha = [\alpha_W | \alpha_R]^T$

dual problem:

$$\max_\alpha [\alpha_W | \alpha_R] 1 - \frac{1}{2} [\alpha_W | \alpha_R] \begin{bmatrix} D_{WW} D_{WR} \\ D_{RW} D_{RR} \end{bmatrix} \begin{bmatrix} \alpha_W \\ \alpha_R \end{bmatrix}$$

subject to

$[\alpha_W | \alpha_R] y = 0$ and $0 \leqq \alpha \leqq c$.

the reduced problem:

treat α_W as variables and α_R as constraints, that is,

$$\max_{\alpha_W} \alpha_W^T (1 - D_{WR} \alpha_R) - \frac{1}{2} \alpha_W^T D_{WW} \alpha_W$$

subject to

$\alpha_W^T y_W = - \alpha_R^T y_R$ and $0 \leqq \alpha_W \leqq c$

where $y = [y_W | y_R]$.

-> no theoretical proof the convergence of this method has been given, but in practice this method works very well.

algorithm:

Given training set S

Select an arbitrary working set α_W.

Repeat

solve the optimization problem on α_W with α_R as constraints.

select a new working set not satisfying

Kuhn-Tucker conditions.

until stopping criterion satisfied.

Return α.

- References

. S/W packages:

LOQO (Princeton Univ., http://www.princeton.edu/rvdv)

MATLAB optimization package (QP solver)

SVMFu (MIT, http://fpn.mit.edu/SvmFu)

LIBSVM, BSVM (NTU, http://www.csie.ntu.edu.tw/~cjlin)

SVM-Light (http://svmlight.joachims.org)

Scikit-learn: Machine Learning in Python

(JMLR, vol. 12: 2825—2830, 2011)

. general information:

Support Vector Machines (http://www.support-vector.net)