2009年11月27日
下午 03:07

$$\arg\min_{w,b} \frac{1}{2}\|w\|^2$$

subject to $t_n(w^T\phi(x_n)+b) \geq 1 \quad n=1,\ldots,N$

quadratic programming with linear inequality constraints

How to analyze and solve such an optimization problem?

Primal optimization problem

Given functions f, h_n, $n=1,\ldots,N$, defined on a domain $\Omega \subseteq \mathbb{R}^D$

minimize $f(w)$, $w \in \Omega$,

subject to $h_n(w)=0$, $n=1,\ldots,N$

whe $f(w)$ is called the objective function, and the equalitis regarding h_n are called equality constraints

The Lagrangian function is defined as

$$L(w, a) = f(w) + \sum_n^N a_n h_n(w)$$

a_n are called the Lagrange multipliers.

Lagrange Theorem: A necessary condition for a normal point w^* to be a minimum of $f(w)$ subject to $h_n(w)=0$, $n=1,\ldots,N$ is

$$\frac{\partial L(w^*, a^*)}{\partial w} = 0 \quad \text{and} \quad \underbrace{\frac{\partial L(w^*, a^*)}{\partial a} = 0}_{\text{original constraints}}$$

for some value a^*.

(Note: the conditions are sufficient if $f(w)$ is convex)

2009年11月27日
下午 03:44

how about inequality constraints?

minimize $f(w)$ $w \in \Omega$

subject to $g_n(w) \leq 0$, $n=1,\ldots,N$

the Lagrangian function is $L(w, a) = f(w) + \sum_{n=1}^N a_n g_n(w)$

The Lagrangian dual problem of the primal problem is

maximize $\theta(a)$

subject to $a_n \geq 0$, $n=1,\ldots,N$

where $\theta(a) = \inf_{w\in\Omega} L(w, a)$.

$\boxed{\text{Kuhn-Tucker Theorem}}$

Given an optimization problem with convex domain $\Omega \subseteq \mathbb{R}^D$

minimize $f(w)$, $w \in \Omega$

subject to $g_n(w) \leq 0$, $n=1,\ldots,N$

with $f \in C^1$ convex and g_n affine, the necessary and sufficient conditions for a normal point w^* to be an optimum are the existence of a_n such that

$$\frac{\partial L(w^*, a^*)}{\partial w} = 0$$

(KKT complementary conditions) $a_n^* g_n(w^*) = 0$, $n=1,\ldots,N$

$g_n(w^*) \leq 0$, $n=1,\ldots,N$

$a_n^* \geq 0$, $n=1,\ldots,N$

A simple Example

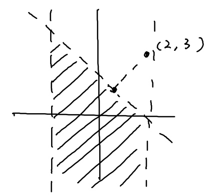
2009年11月27日
下午 03:57

minimize $(x_1-2)^2+(x_2-3)^2$

subject to $x_1+x_2-2\le 0$

$x_1-2\le 0$

$-x_1-2\le 0$



$$L(x,a) = (x_1-2)^2+(x_2-3)^2 + a_1(x_1+x_2-2) + a_2(x_1-2) + a_3(-x_1-2)$$

KKT : $\frac{\partial L}{\partial x_1} = 2(x_1-2)+a_1+a_2-a_3 = 0$

$\frac{\partial L}{\partial x_2} = 2(x_2-3)+a_1 = 0$

$a_1, a_2 \geqslant 0$ $\quad x_1+x_2-2\le 0,\ x_1-2\le 0,\ -x_1-2\le 0$

$a_1(x_1+x_2-2)=0$

$a_2(x_1-2)=0$

$a_3(-x_1-2)=0$

case 1 : no constraint is tight

$x_1+x_2-2<0 \quad x_1-2<0,\ -x_1-2<0$

by KKT $a_1=a_2=a_3=0 \Rightarrow 2(x_1-2)=0$ and $2(x_2-3)=0$

we get $(x_1,x_2)=(2,3)$ not a feasible solution ✗

case 2 : only $x_1-2=0$ is tight

by KKT $a_1=0 \Rightarrow 2(x_2-3)=0 \Rightarrow (2,3)$ not a solution

case 3: only $x_1+x_2-2=0$ is tight

by KKT $a_2=0$, $a_3=0$, $x_1+x_2-2=0$ $(a_1\geqslant 0)$

$\Rightarrow 2(x_1-2)+a_1=0,\ x_1+x_2-2=0,$ and $2(x_2-3)+a_1=0$

$\Rightarrow (\frac{1}{2},\frac{3}{2})$ is a global solution $(a_1=3)$

Apply Lagrange multipliers to large margin optimization

2009年11月27日
下午 11:09

Lagrangian function

$$L(w,b,a) = \frac{1}{2}\|w\|^2 - \sum_{n=1}^{N} a_n\{t_n(w^T\phi(x_n)+b)-1\}$$

maximize w.r.t. a, minimize w.r.t. w, b

$\frac{\partial L}{\partial w}=0 \Rightarrow w = \sum_{n=1}^{N} a_n t_n \phi(x_n)$

$\frac{\partial L}{\partial b}=0 \Rightarrow 0 = \sum_{n=1}^{N} a_n t_n$

substitute w and b in $L(w,b,a)$

(maximize) $\tilde{L}(a) = \sum_{n=1}^{N} a_n - \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} a_n a_m t_n t_m k(x_n,x_m)$

$a_n\geqslant 0$, $\sum_{n=1}^{N} a_n t_n = 0$

k is positive definite $\Rightarrow \tilde{L}(a)$ is bounded below

$$\boxed{y(x) = \sum_{n=1}^{N} a_n t_n k(x,x_n) + b}$$

KKT $a_n\geqslant 0$

$t_n(w^T\phi(x_n)+b)-1\geqslant 0 \Rightarrow t_n y(x_n)-1\geqslant 0$

$a_n\{t_n(w^T\phi(x_n)+b)-1\}=0 \Rightarrow a_n\{t_n y(x_n)-1\}=0$

either $a_n=0$ or $t_n y(x_n)-1=0$

$a_n\neq 0$ support vectors $\Rightarrow t_n y(x_n)=1 \Rightarrow$ $\boxed{\text{on maximum margin hyperplane}}$

2009年11月29日
下午 03:51

Suppose we have solved for $\vec{a}$

x_n satisfies $t_n y(x_n) = 1$, x_n is a support vector

$$\hookrightarrow \quad t_n \left(\sum_{m \in S} a_m t_m k(x_n, x_m) + b \right) = 1$$

S : the index set of support vectors

To decide b , multiply the above equation by t_n

$$t_n^2 \left(\sum_{m \in S} a_m t_m k(x_n, x_m) + b \right) = t_n \qquad \| \ t_n^2 = 1$$

$$\Rightarrow \quad \sum_{m \in S} a_m t_m k(x_n, x_m) + b = t_n \qquad \| \text{ sum up all } n \in S$$

$$\Rightarrow \quad \sum_{n \in S} \left\{ \sum_{m \in S} a_m t_m k(x_n, x_m) \right\} + N_S b = \sum_{n \in S} t_n \qquad \| \ N_S = |S|$$

$$\Rightarrow \quad b = \frac{1}{N_S} \sum_{n \in S} \left(t_n - \sum_{m \in S} a_m t_m k(x_n, x_m) \right)$$

so $y(x) = \sum_{n=1}^{N} a_n t_n k(x, x_n) + b$ is decided

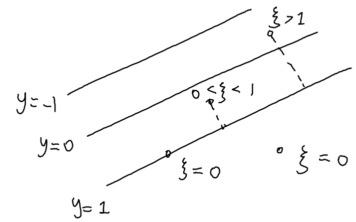
given a new input x, we can use $y(x)$ to predict the class of x according to $\operatorname{sign}(y(x))$

So far we assumed the training data are linearly separable.
What if the data are not linearly separable?

2009年11月29日
下午 04:04

Soft Margin Optimization

Slack variables $\xi_n \geq 0 \quad n = 1, \dots, N$



$\xi_n > 1$
misclassification

the constraints become $t_n y(x_n) \geq 1 - \xi_n \quad n = 1, \dots, N$

The is called "soft margin" optimization.
Some of the training data are allowed to be misclassified.

$$\text{minimize} \quad C \sum_{n=1}^{N} \xi_n + \frac{1}{2} \|w\|^2 \qquad (C > 0)$$

$$\text{subject to} \quad \xi_n \geq 0, \quad t_n y(x_n) \geq 1 - \xi_n, \quad n = 1, \dots, N$$

$\|$ $\xi_n > 1 \Rightarrow$ misclassified, the penalty term $\sum_n \xi_n$ is an upper bound on the number of misclassified points

The Lagrangian function is

$$L(w, b, \xi, \vec{a}, \mu) = \frac{1}{2}\|w\|^2 + C \sum_{n=1}^{N} \xi_n - \sum_{n=1}^{N} a_n \{ t_n y(x_n) - 1 + \xi_n \} - \sum_{n=1}^{N} \mu_n \xi_n \qquad n = 1, \dots, N$$

$\|$ μ_n, a_n are Lagrange multipliers

2009年11月29日
下午 04:15

$$\frac{\partial L}{\partial w} = 0 \Rightarrow w = \sum_{n=1}^{N} a_n t_n \phi(x_n)$$

$$\frac{\partial L}{\partial b} = 0 \Rightarrow \sum_{n=1}^{N} a_n t_n = 0$$

$$\frac{\partial L}{\partial \xi_n} = 0 \Rightarrow a_n = C - \mu_n$$

KKT

$$a_n \geq 0, \quad \mu_n \geq 0, \quad \xi_n \geq 0$$

$$t_n y(x_n) - 1 + \xi_n \geq 0$$

$$a_n (t_n y(x_n) - 1 + \xi_n) = 0$$

$$\mu_n \xi_n = 0 \qquad n = 1, \dots, N$$

$$\tilde{L}(a) = \sum_{n=1}^{N} a_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} a_n a_m t_n t_m k(x_n, x_m)$$

$$a_n = C - \mu_n, \quad \mu_n \geq 0 \Rightarrow 0 \leq a_n \leq C$$

maximize $\tilde{L}(a)$

subject to $0 \leq a_n \leq C$,

$\sum_{n=1}^{N} a_n t_n = 0, \quad n = 1, \dots, N.$

The optimization problem can be solved efficiently by the Sequential Minimal Optimization (SMO) algorithm, which optimizes two points at each iteration.
The optimization problem for two data points has an analytical solution. Another advantage of SMO is that it does not need to store the kernel matrix, since no matrix operations are involved. SMO is easy to implement (~100 lines of code)

2009年11月29日
下午 04:42

Soft Margin SVM

$a_n = 0$ useless

$a_n > 0$ support vectors $t_n y(x_n) = 1 - \xi_n$

$\hookrightarrow \begin{cases} a_n < C \Rightarrow \mu_n > 0 \ (a_n = C - \mu_n) \Rightarrow \xi_n = 0 \ (\mu_n \xi_n = 0) \Rightarrow \text{on margin} \\ a_n = C \Rightarrow \mu_n = 0 \Rightarrow \xi_n > 0 \Rightarrow \text{margin error} \end{cases}$

To decide b

for those support vectors x_n with $0 < a_n < C \Rightarrow \xi_n = 0$

$t_n y(x_n) - 1 = 0$

$$t_n \left(\sum_{m \in S} a_m t_m k(x_n, x_m) + b \right) = 1$$

$$y(x) = \sum_{m \in S} a_m t_m k(x, x_m) + b$$

$$b = \frac{1}{N_M} \sum_{n \in M} \left(t_n - \sum_{m \in S} a_m t_m k(x_n, x_m) \right)$$

$$M = \{ n \mid 0 < a_n < C \}$$

$\boxed{\nu\text{-SVM}}$

minimize $\frac{1}{2}\|w\|^2 - \nu\rho + \frac{1}{N}\sum_n \xi_n$

subject to $t_n (w^T \phi(x_n) + b) \geq \rho - \xi_n$

$\xi_n \geq 0, \quad \rho \geq 0$ $\qquad \frac{2\rho}{\|w\|}$ margin

$$L(w, \xi, b, \rho, a, \mu, \delta)$$
$$= \frac{1}{2}\|w\|^2 - \nu\rho + \frac{1}{N}\sum_n \xi_n - \sum_n a_n \{ t_n (w^T \phi(x_n) + b) - \rho + \xi_n \}$$
$$- \sum_n \mu_n \xi_n - \delta\rho$$

$a_n, \mu_n, \delta \geq 0 \qquad n = 1, \dots, N$

2009年11月29日
下午 06:13

$$\frac{\partial L}{\partial w} = 0 \Rightarrow w = \sum_n a_n t_n \phi(x_n)$$

$$\frac{\partial L}{\partial \xi_n} = 0 \Rightarrow a_n + \mu_n = \frac{1}{N}$$

$$\frac{\partial L}{\partial b} = 0 \Rightarrow 0 = \sum_n a_n t_n$$

$$\frac{\partial L}{\partial \rho} = 0 \Rightarrow \sum_n a_n - \delta = \nu$$

$$\tilde{L}(a) = -\frac{1}{2} \sum_n \sum_m a_n a_m t_n t_m k(x_n, x_m)$$

subject to $0 \le a_n \le \frac{1}{N}$

$$\sum_n a_n t_n = 0$$

$$\sum_n a_n \geq \nu$$

by KKT $\rho\delta = 0$, $\rho > 0 \Rightarrow \delta = 0 \Rightarrow \sum_n a_n = \nu$

① for those $\xi_n > 0$ (margin error) $\Rightarrow a_n = \frac{1}{N}$

$$\frac{\#\text{ of margin errors}}{N} \le \sum_{n \in \text{margin error}} a_n + \sum_{n \in \text{remaining SVs}} a_n = \sum_{n=1}^{N} a_n = \nu$$

so there are at most ν fraction of training data with $\xi_n > 0$

② $$\frac{\#\text{ of SVs}}{N} \geq \sum_n a_n = \nu$$

ν is the lower bound of the fraction of support vectors to the training data

2009年11月29日
下午 06:25

To decide b

by KKT
for x_n with $0 < a_n < \frac{1}{N}$, $t_n y(x_n) - \rho = 0$
(since $\xi_n = 0$)

$$t_n \left(\sum_{m=1}^{N} a_m t_m k(x_n, x_m) + b \right) - \rho = 0$$

consider two index set S_+, S_-

$$S_+ = \{ n \mid 0 < a_n < \tfrac{1}{N},\ t_n = +1 \}$$

$$S_- = \{ n \mid 0 < a_n < \tfrac{1}{N},\ t_n = -1 \}$$

$$\begin{cases} +\sum_{n \in S_+} \sum_{m=1}^{N} a_m t_m k(x_n, x_m) + b|S_+| - \rho|S_+| = 0 \\ -\sum_{n \in S_-} \sum_{m=1}^{N} a_m t_m k(x_n, x_m) - b|S_-| - \rho|S_-| = 0 \end{cases}$$

assume $|S_+| = |S_-| = N_S$

$$\begin{cases} b = -\frac{1}{2N_S} \sum_{n \in S_+ \cup S_-} \sum_{m=1}^{N} a_m t_m k(x_n, x_m) \\ \rho = \frac{1}{2N_S} \left(\sum_{n \in S_+} \sum_{m=1}^{N} a_m t_m k(x_n, x_m) - \sum_{n \in S_-} \sum_{m=1}^{N} a_m t_m k(x_n, x_m) \right) \end{cases}$$

$$y(x) = \sum_{m=1}^{N} a_m t_m k(x, x_m) + b$$

2009年11月29日
下午 06:55

MULTICLASS SVMs

Fundamentally SVMs are two-class classifiers.
Various approaches to applying SVMs to multiclass classification:

① one-versus-the-rest (one-against-all)

training K separate SVMs

prediction by $y(x) = \max_k y_k(x)$

drawbacks: ① $y_k(x)$ may have different scales
② imbalanced training set
negative size $\gg$ positive size

② one-versus-one (one-against-one)

requires more training and test time

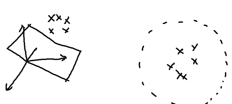
training $\frac{K(K-1)}{2}$ SVMs

prediction by majority voting

a variant for speeding up test time : DAGSVM

③ error-correcting-output-code
(ECOC)

④ single-class SVM



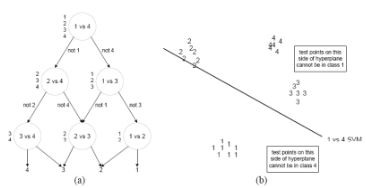


Figure 1: (a) The decision DAG for finding the best class out of four classes. The equivalent list state for each node is shown next to that node. (b) A diagram of the input space of a four-class problem. A 1-v-1 SVM can only exclude one class from consideration.