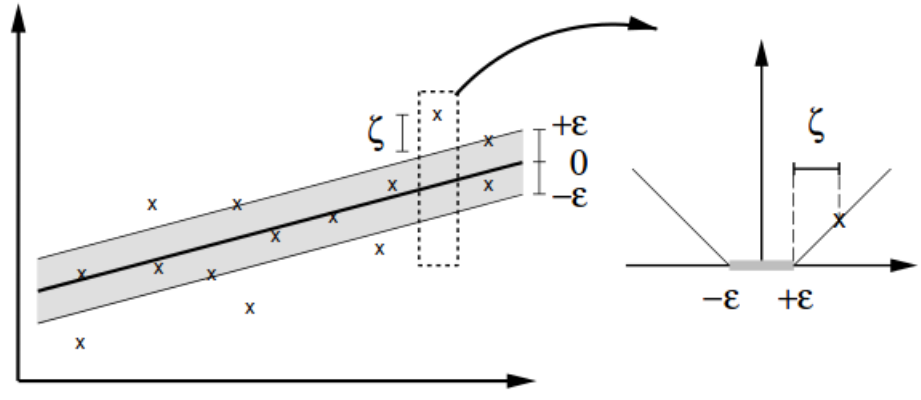
SVR原理浅析



方敏

驰声研发部 **2018.07**

主要内容

- 目标函数定义
- 对偶问题与kkt条件

目标函数定义

线性**svr**

$$
\begin{aligned}
&f(x) = w * x + b \\
&w \in R^n; b \in R
\end{aligned}
$$

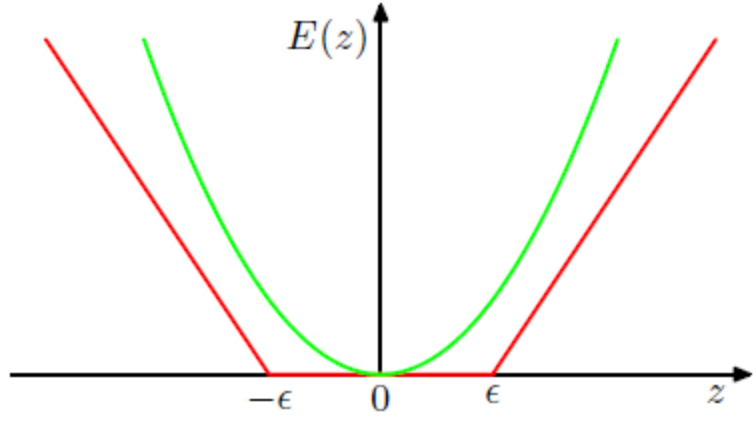
优化目标

$$
\begin{aligned}
min \quad & \frac{1}{2} \parallel w \parallel^2 \\
s.t. \quad & \begin{cases} t - w * x - b \leq \epsilon \\ w * x + b - t \leq \epsilon \end{cases}
\end{aligned}
$$

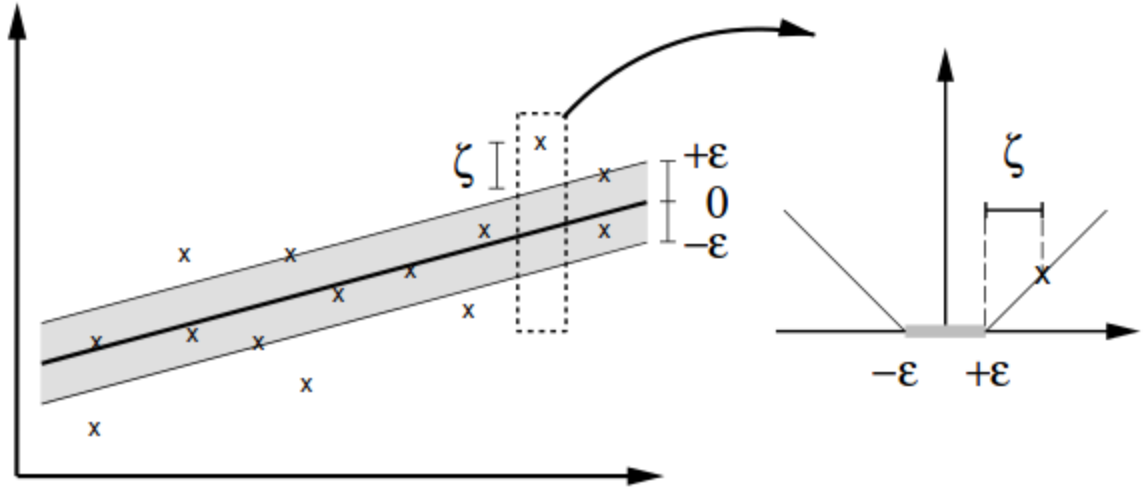
损失函数

$$E_\epsilon(y(x)-t)=\begin{cases}0 & |w*x+b-t|<\epsilon \\ |w*x+b-t|-\epsilon & other\end{cases}$$

Plot of an ϵ-insensitive error function (in red) in which the error increases linearly with distance beyond the insensitive region. Also shown for comparison is the quadratic error function (in green).



线性SVR



目标函数定义

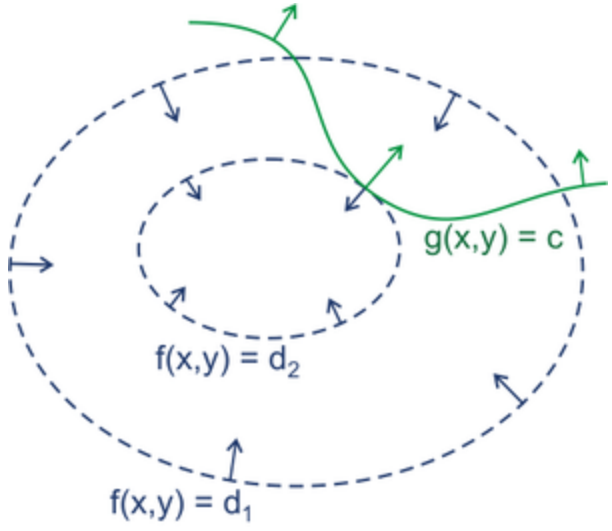
$$
\begin{aligned}
min \quad & \frac{1}{2} \parallel w \parallel^2 + C \sum_{n=1}^{N} (\xi_n + \hat{\xi}_n) \\
s.t. \quad & \begin{cases} t - w * x - b - \xi_n \leq \epsilon \\ w * x + b - t - \hat{\xi}_n \leq \epsilon \\ \xi_n \geq 0 \\ \hat{\xi}_n \geq 0 \end{cases}
\end{aligned}
$$

对偶问题与kkt条件

对偶问题

$$\begin{aligned} min_x f(x) \quad &= min_x max_u L(x,u) \\ &= max_u min_x L(x,u) \end{aligned}$$

拉格朗日乘子法与kkt条件



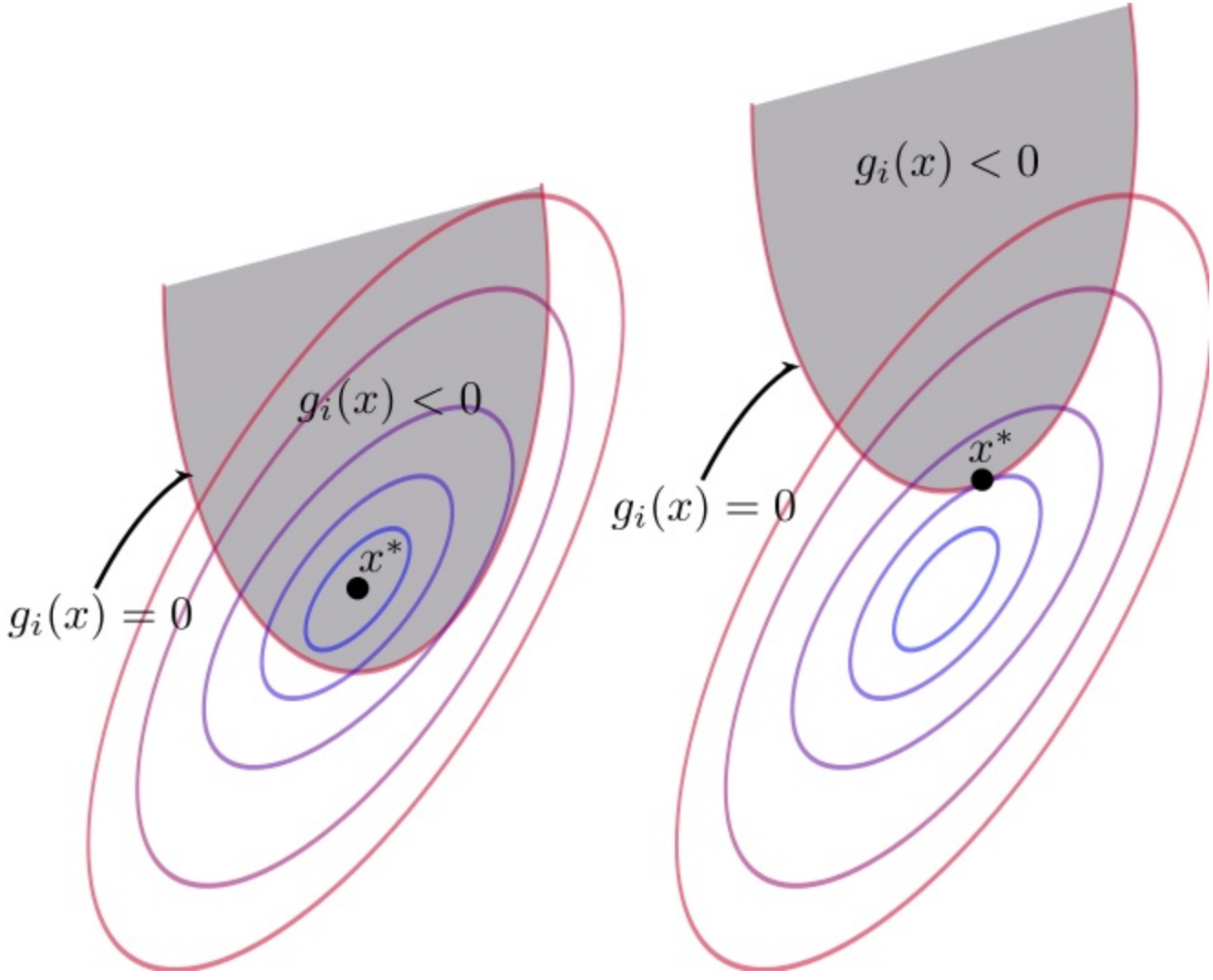
$$\begin{cases} min & f(x,y) \\ s.t. & g(x,y) = c \end{cases} \equiv minF(x) = f(x,y) + \lambda(g(x,y) - c)$$

kkt是拉格朗日乘子法的泛化

$$
\begin{cases}
L(x,\lambda,u) = f(x) + \sum_{i=1}^{n} \lambda_i h_i(x) + \sum_{k=1}^{q} u_k g_k(x) \\
\lambda_i \neq 0 \\
h_i(x) = 0 \\
u_k \geq 0 \\
g_k(x) \leq 0
\end{cases}
\Rightarrow
$$

$$
\begin{cases}
min_x max_u L(x,\lambda,u) & = max_u min_x L(x,\lambda,u) \\
& = min_x f(x) = f(x^\star) \\
& u_k g_k(x^\star) = 0 \\
& \frac{\partial L(x,\lambda,u)}{\partial x}|_{x=x^\star} = 0
\end{cases}
$$

Karush-Kuhn-Tucker(KKT)



SVR的拉格朗日函数

$$
\begin{aligned}
L = & \frac{1}{2} \parallel w \parallel^2 + C \sum_{n=1}^{N} (\xi_n + \hat{\xi}_n) \\
& - \sum_{n=1}^{N} a_n (\epsilon + \xi + y_n - t_n) \\
& - \sum_{n=1}^{N} \hat{a_n} (\epsilon + \hat{\xi}_n - y_n + t_n)
\end{aligned}
$$

```
[mf34@paris Pronounciation]$ /aiwork6/sy123/mytools/libsvm-3.11/64bin/svm-train
Usage: svm-train [options] training_set_file [model_file]
options:
-s svm_type : set type of SVM (default 0)
        0 -- C-SVC
        1 -- nu-SVC
        2 -- one-class SVM
        3 -- epsilon-SVR
        4 -- nu-SVR
-t kernel_type : set type of kernel function (default 2)
        0 -- linear: u'*v
        1 -- polynomial: (gamma*u'*v + coef0)^degree
        2 -- radial basis function: exp(-gamma*|u-v|^2)
        3 -- sigmoid: tanh(gamma*u'*v + coef0)
        4 -- precomputed kernel (kernel values in training_set_file)
-d degree : set degree in kernel function (default 3)
-g gamma : set gamma in kernel function (default 1/num_features)
-r coef0 : set coef0 in kernel function (default 0)
-c cost : set the parameter C of C-SVC, epsilon-SVR, and nu-SVR (default 1)
-n nu : set the parameter nu of nu-SVC, one-class SVM, and nu-SVR (default 0.5)
-p epsilon : set the epsilon in loss function of epsilon-SVR (default 0.1)
-m cachesize : set cache memory size in MB (default 100)
-e epsilon : set tolerance of termination criterion (default 0.001)
-h shrinking : whether to use the shrinking heuristics, 0 or 1 (default 1)
-b probability_estimates : whether to train a SVC or SVR model for probability estimates, 0 or 1 (default 0)
-wi weight : set the parameter C of class i to weight*C, for C-SVC (default 1)
-v n: n-fold cross validation mode
-q : quiet mode (no outputs)
```

参考文献

- 【机器学习详解】SVM解回归问题
- A Tutorial on Support Vector Regression
- 拉格朗日乘子法和KKT条件

over

thank you!!!