Variants of Support Vector Machines

Ben Dai

1 Weighted support vector machine

Given a training dataset of n points of the form $(\boldsymbol{x}_i, y_i)_{i=1}^n$, where $y_i = \pm 1$ which indicates the class of the instance $\boldsymbol{x}_i \in \mathbb{R}^d$. Then the support vector machine [1] formulates as

$$\min_{\boldsymbol{\beta}} \sum_{i=1}^n C_i (1 - y_i \boldsymbol{x}_i^T \boldsymbol{\beta})_+ + \frac{1}{2} \|\boldsymbol{\beta}\|_2^2, \tag{1}$$

where C_i is a tuning parameter controlling the trade-off between the training loss and magnitude of the parameters. After introducing some slack variables,

$$\begin{aligned} &\min_{\boldsymbol{\beta}} \sum_{i=1}^n C_i \xi_i + \frac{1}{2} \|\boldsymbol{\beta}\|_2^2 \\ \text{subj to} \quad & \xi_i \geq 0, \ y_i \boldsymbol{x}_i^T \boldsymbol{\beta}_i \geq 1 - \xi_i, \text{for } i = 1, \cdots, n. \end{aligned} \tag{2}$$

Here (2) is a quadratic with linear inequality constrains, we convert it to the dual version by using Lagrange multipliers. Specifically, the Lagrange function is

$$L_P = \frac{1}{2} \|\boldsymbol{\beta}\|_2^2 + \sum_{i=1}^n C_i \xi_i - \sum_{i=1}^n \alpha_i \big(y_i \boldsymbol{x}_i^T \boldsymbol{\beta} - (1 - \xi_i) \big) - \sum_{i=1}^n \mu_i \xi_i.$$

Taking the derivatives with respect to $\boldsymbol{\beta}$ and ξ, we get

$$\boldsymbol{\beta} = \sum_{i=1}^n \alpha_i y_i \boldsymbol{x}_i; \quad \alpha_i = C_i - \mu_i;$$

and the nonnegative constrains α_i, μ_i and $\xi_i \geq 0$. Then the dual objective function is

$$\min_{\boldsymbol{\alpha}} \frac{1}{2} \boldsymbol{\alpha}^T \boldsymbol{Q} \boldsymbol{\alpha} - \boldsymbol{e}^T \boldsymbol{\alpha}; \quad \text{subj to} \quad 0 \leq \boldsymbol{\alpha} \leq C_i, \tag{3}$$

where $\boldsymbol{Q}_{ij} = y_i y_j \boldsymbol{x}_i^T \boldsymbol{x}_j$. If $n > d$, we update $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ simultaneously to deduce both time and memory complexity [2]. The i-th coordinate subproblem yields that

$$\delta^* = \max\Big(-\alpha_i, \min\big(C_i - \alpha_i, \frac{1 - y_i \boldsymbol{\beta}^T \boldsymbol{x}_i}{\boldsymbol{Q}_{ii}}\big)\Big); \quad \alpha_i \leftarrow \alpha_i + \delta^*; \quad \boldsymbol{\beta} \leftarrow \boldsymbol{\beta} + \delta^* y_i \boldsymbol{x}_i. \tag{4}$$

If $d \geq n$, then update the dual variable is less time consuming , the i-th coordinate subproblem yields that

$$\delta_i^* = \max\Big(-\alpha_i, \min\big(C_i - \alpha_i, \frac{1 - (\boldsymbol{Q}\boldsymbol{\alpha})_i}{\boldsymbol{Q}_{ii}}\big)\Big); \quad \alpha_i \leftarrow \alpha_i + \delta^*.$$

In following variant models, we focus on the coordinate descent of (4).

2 Drifted support vector machines

The drifted support vector machine is a SVM with fixed intercept, which formulates as

$$\min_{\boldsymbol{\beta}} \sum_{i=1}^n C_i (1 - y_i(\boldsymbol{x}_i^T \boldsymbol{\beta} + d_i))_+ + \frac{1}{2}\|\boldsymbol{\beta}\|_2^2,$$

where C_i is a tuning parameter controlling the trade-off between the training loss and magnitude of the parameters. After introducing some slack variables,

$$\begin{aligned} &\min_{\boldsymbol{\beta}} \sum_{i=1}^n C_i \xi_i + \frac{1}{2}\|\boldsymbol{\beta}\|_2^2 \\ &\text{subj to} \quad \xi_i \geq 0,\ y_i(\boldsymbol{x}_i^T \boldsymbol{\beta}_i + d_i) \geq 1 - \xi_i, \text{for } i = 1, \cdots, n. \end{aligned}$$

Similarly, we convert it to the dual version by using Lagrange multipliers,

$$L_P = \frac{1}{2}\|\boldsymbol{\beta}\|_2^2 + \sum_{i=1}^n C_i \xi_i - \sum_{i=1}^n \alpha_i \big(y_i(\boldsymbol{x}_i^T \boldsymbol{\beta} + d_i) - (1 - \xi_i)\big) - \sum_{i=1}^n \mu_i \xi_i.$$

Taking the derivatives with respect to $\boldsymbol{\beta}$ and ξ, we get

$$\boldsymbol{\beta} = \sum_{i=1}^n \alpha_i y_i \boldsymbol{x}_i; \quad \alpha_i = C_i - \mu_i;$$

and the nonnegative constrains α_i, μ_i and $\xi_i \geq 0$. Then the dual objective function is

$$\min_{\boldsymbol{\alpha}} \frac{1}{2}\boldsymbol{\alpha}^T \boldsymbol{Q} \boldsymbol{\alpha} - (\boldsymbol{e} - \bar{\boldsymbol{d}})^T \boldsymbol{\alpha}; \quad \text{subj to} \quad 0 \leq \boldsymbol{\alpha} \leq C_i,$$

where $\bar{\boldsymbol{d}} = (y_1 d_1, \cdots, y_n d_n)^T$, $\boldsymbol{Q}_{ij} = y_i y_j \boldsymbol{x}_i^T \boldsymbol{x}_j$. Then we update $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ simultaneously. The i-th coordinate subproblem yields that

$$\delta^* = \max\Big(-\alpha_i, \min\big(C_i - \alpha_i, \frac{1 - \bar{d}_i - y_i \boldsymbol{\beta}^T \boldsymbol{x}_i}{\boldsymbol{Q}_{ii}}\big)\Big); \quad \alpha_i \leftarrow \alpha_i + \delta^*; \quad \boldsymbol{\beta} \leftarrow \boldsymbol{\beta} + \delta^* y_i \boldsymbol{x}_i.$$

3 Nonnegative drifted support vector machines

The nonnegative drifted support vector machine is a drifted SVM with nonnegative constrains in parameters, that is

$$\min_{\boldsymbol{\beta}} \sum_{i=1}^n C_i (1 - y_i(\boldsymbol{x}_i^T \boldsymbol{\beta} + d_i))_+ + \frac{1}{2}\|\boldsymbol{\beta}\|_2^2, \quad \text{subj to} \quad \beta_j \geq 0.$$

After introducing some slack variables,

$$\begin{aligned} &\min_{\boldsymbol{\beta}} \sum_{i=1}^n C_i \xi_i + \frac{1}{2}\|\boldsymbol{\beta}\|_2^2 \\ &\text{subj to} \quad \beta_j \geq 0, \xi_i \geq 0,\ y_i(\boldsymbol{x}_i^T \boldsymbol{\beta}_i + d_i) \geq 1 - \xi_i, \text{for } i = 1, \cdots, n; j = 1, \cdots, d. \end{aligned} \tag{5}$$

The dual version by using Lagrange multipliers is

$$L_P = \frac{1}{2}\|\boldsymbol{\beta}\|_2^2 + \sum_{i=1}^n C_i \xi_i - \sum_{i=1}^n \alpha_i \big(y_i(\boldsymbol{x}_i^T \boldsymbol{\beta} + d_i) - (1 - \xi_i)\big) - \sum_{i=1}^n \mu_i \xi_i - \sum_{j=1}^d \rho_j \beta_j.$$

Taking the derivatives with respect to $\boldsymbol{\beta}$ and ξ, we get

$$\boldsymbol{\beta} = \sum_{i=1}^n \alpha_i y_i \boldsymbol{x}_i + \boldsymbol{\rho} = \bar{\boldsymbol{X}}^T \boldsymbol{\alpha} + \boldsymbol{\rho}; \quad \alpha_i = C_i - \mu_i;$$

and the nonnegative constrains α_i, μ_i, ρ_j and $\xi_i \geq 0$. Then the dual objective function is

$$\min_{\boldsymbol{\alpha}, \boldsymbol{\rho}} \frac{1}{2}\boldsymbol{\alpha}^T \boldsymbol{Q} \boldsymbol{\alpha} + \frac{1}{2}\boldsymbol{\rho}^T \boldsymbol{\rho} + \boldsymbol{\alpha}^T \bar{\boldsymbol{X}} \boldsymbol{\rho} - (\boldsymbol{e} - \bar{\boldsymbol{d}})^T \boldsymbol{\alpha}; \quad \text{subj to} \quad 0 \leq \alpha_i \leq C_i,\ \rho_j \geq 0, \tag{6}$$

where $\bar{\boldsymbol{d}} = (y_1 d_1, \cdots, y_n d_n)^T$, $\boldsymbol{Q}_{ij} = y_i y_j \boldsymbol{x}_i^T \boldsymbol{x}_j$ and $\bar{\boldsymbol{X}}$ is the matrix with i-th row being $y_i \boldsymbol{x}_i$. Then we update $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ simultaneously to deduce both time and memory complexity. Taking the derivative of the i-th coordinate subproblem yields that

$$(\boldsymbol{Q}\boldsymbol{\alpha})_i + \boldsymbol{Q}_{ii}\delta_\alpha - (1 - \bar{d}_i) + y_i \boldsymbol{x}_i^T \boldsymbol{\rho} = 0; \quad \rho_i + \delta_\rho + (\bar{\boldsymbol{X}}^T \boldsymbol{\alpha})_i = 0.$$

Then, we have

$$\delta_\alpha^* = \max\Big(-\alpha_i, \min\big(C_i - \alpha_i, \frac{1 - \bar{d}_i - y_i \boldsymbol{\beta}^T \boldsymbol{x}_i}{\boldsymbol{Q}_{ii}}\big)\Big);\ \alpha_i \leftarrow \alpha_i + \delta_\alpha^*;\ \boldsymbol{\beta} \leftarrow \boldsymbol{\beta} + \delta_\alpha^* y_i \boldsymbol{x}_i,$$
$$\delta_\rho^* = \max\Big(-\rho_i, -\rho_i - (\bar{\boldsymbol{X}}^T \boldsymbol{\alpha})_i\Big) = \max(-\rho_i, -\beta_i); \quad \rho_i \leftarrow \rho_i + \delta_\rho^*; \quad \beta_i \leftarrow \beta_i + \delta_\rho^*.$$

References

[1] Cortes, C., and Vapnik, V. (1995). Support-vector networks. *Machine Learning*, **20**(3), 273-297.

[2] Fan, R. E., Chang, K. W., Hsieh, C. J., Wang, X. R., and Lin, C. J. (2008). LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, **9**(Aug), 1871-1874.