Quadratic Programming: SVMs as Examples

Apple Zhang

2022 年 11 月 1 日

1 上下界约束的二次规划问题

对于支持向量机

$$\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|_2^2 + c\sum_{i=1}^{n}(1 - y_i\mathbf{w}^T\mathbf{x}_i)_+^p \tag{1}$$

其中 $(\cdot)_+ = \max(0, \cdot)$. 对于 $p = 1, 2$ ，上述问题均可以转化为以下形式的优化问题.

$$\tilde{\boldsymbol{\alpha}} = \arg\min_{\boldsymbol{\alpha}} f(\boldsymbol{\alpha}) = \frac{1}{2}\boldsymbol{\alpha}^T\mathbf{Q}\boldsymbol{\alpha} + \mathbf{h}^T\boldsymbol{\alpha}, \quad 0 \le \alpha_i \le c. \tag{2}$$

其中 $f(\boldsymbol{\alpha})$ 为优化目标函数. 则 (1) 的最优解为

$$\mathbf{w} = \sum_{i=1}^{n} y_i\tilde{\alpha}_i\mathbf{x}_i. \tag{3}$$

1.1 坐标下降法

文献 [1] 提出了采用坐标下降 (Coordinate Descent, CD) 的方法求解问题 (2). 在每个 epoch 中将更新 $\boldsymbol{\alpha}$ 的每个元素且每个元素只更新一次. 当更新第 i 个元素，即 α_i 时，固定其它的元素不变. 假设 α_i 将被更新为 $\alpha_i + d$，则问题转化为

$$\min_{d} f(\boldsymbol{\alpha} + d\mathbf{e}_i), \quad \text{s.t. } 0 \le \alpha_i + d \le c. \tag{4}$$

其中 $\mathbf{e}_i$ 表示只有第 i 个元素为 1 ，其余元素均为 0 的列向量. 化简：

$$\begin{aligned}
& f(\boldsymbol{\alpha} + d\mathbf{e}_i), \\
= & \frac{1}{2}(\boldsymbol{\alpha} + d\mathbf{e}_i)^T\mathbf{Q}(\boldsymbol{\alpha} + d\mathbf{e}_i) + \mathbf{h}^T(\boldsymbol{\alpha} + d\mathbf{e}_i), \\
= & \frac{1}{2}\mathbf{e}_i^T\mathbf{Q}\mathbf{e}_i d^2 + (\mathbf{Q}\boldsymbol{\alpha})_i d + h_i d + f(\boldsymbol{\alpha}), \\
= & \frac{1}{2}Q_{ii}d^2 + [(\mathbf{Q}\boldsymbol{\alpha})_i + h_i]d + \text{const} \\
= & \frac{1}{2}Q_{ii}d^2 + \nabla_i f(\boldsymbol{\alpha})d + \text{const}
\end{aligned}$$

其中 $\nabla_i f(\boldsymbol{\alpha})$ 表示 $f(\boldsymbol{\alpha})$ 对 α_i 的梯度. 即 (4) 等价于

$$\min_d \frac{1}{2}Q_{ii}d^2+\nabla_i f(\boldsymbol{\alpha})d, \quad \text{s.t. } 0\le \alpha_i+d\le c. \tag{5}$$

若不考虑约束，则该问题最优解为

$$d=-\frac{\nabla_i f(\boldsymbol{\alpha})}{Q_{ii}}. \tag{6}$$

在考虑约束时，α_i 的更新不能逃离 $[0,c]$ 的区间，因此得到以下的更新公式：

$$\alpha_i:=\pi_{[0,c]}\left(\alpha_i-\frac{\nabla_i f(\boldsymbol{\alpha})}{Q_{ii}}\right). \tag{7}$$

其中 $\pi_{[0,c]}(\cdot)$ 表示到区间 $[0,c]$ 上的投影，即 $\pi_{[0,c]}(\cdot)=\min(\max(\cdot,0),c)$.

在以上更新中，每次计算 $\nabla_i f(\boldsymbol{\alpha})$ 都涉及到矩阵 $\mathbf{Q}$ 的计算，为了降低计算量，[1] 采用了“$\mathbf{w}$ 技巧”，注意到 [1]

$$(\mathbf{Q}\boldsymbol{\alpha})_i=\sum_{j=1}^n y_iy_j\mathbf{x}_j^T\mathbf{x}_i\alpha_j=\sum_{j=1}^n(\alpha_jy_j\mathbf{x}_j)^T\mathbf{x}_i=\mathbf{w}^T\mathbf{x}_i. \tag{8}$$

因此 $\nabla_i f(\boldsymbol{\alpha})=\mathbf{w}^T\mathbf{x}_i+h_i$. 从而不需要存储整个矩阵 $\mathbf{Q}$，只需存储其所有对角线元素即可进行计算. 而在每一次更新 α_i 后，我们也需要更新 $\mathbf{w}$ 为

$$\mathbf{w}:=\mathbf{w}+(\alpha_i^{new}-\alpha_i^{old})y_i\mathbf{x}_i. \tag{9}$$

Example: Inverse Free TWSVM.

考虑孪生支持向量机，

$$\min_{\mathbf{w}_+,\boldsymbol{\xi}} \frac{1}{2}\|\mathbf{A}^T\mathbf{w}_+\|_2^2+c\mathbf{1}^T\boldsymbol{\xi}+\frac{r}{2}\|\mathbf{w}_+\|_2^2, \quad \text{s.t. } \mathbf{1}-\boldsymbol{\xi}+\mathbf{B}^T\mathbf{w}_+\le\mathbf{0}, \quad \boldsymbol{\xi}\ge 0. \tag{10}$$

$$\min_{\mathbf{w}_-,\boldsymbol{\eta}} \frac{1}{2}\|\mathbf{B}^T\mathbf{w}_-\|_2^2+c\mathbf{1}^T\boldsymbol{\eta}+\frac{r}{2}\|\mathbf{w}_-\|_2^2, \quad \text{s.t. } \mathbf{1}-\boldsymbol{\eta}-\mathbf{A}^T\mathbf{w}_-\le\mathbf{0}, \quad \boldsymbol{\eta}\ge 0. \tag{11}$$

其中 $\mathbf{A},\mathbf{B}$ 为正负类矩阵：$\mathbf{A}=[\mathbf{x}_1^+,\cdots,\mathbf{x}_{n_A}^+]$, $\mathbf{B}=[\mathbf{x}_1^-,\cdots,\mathbf{x}_{n_B}^-]$. 两个优化问题的解法相同，所以下面只讨论第一个问题的求解. IF-TWSVM (Inverse Free TWSVM) 通过改写目标函数，使得求解过程不会涉及求逆：

$$\min_{\mathbf{w}_+} \frac{1}{2}\|\mathbf{t}\|_2^2+c\mathbf{1}^T\boldsymbol{\xi}+\frac{r}{2}\|\mathbf{w}_+\|_2^2, \quad \text{s.t. } \mathbf{1}-\boldsymbol{\xi}+\mathbf{B}^T\mathbf{w}_+\le\mathbf{0}, \quad \boldsymbol{\xi}\ge 0, \quad \mathbf{A}^T\mathbf{w}_+=\mathbf{t}. \tag{12}$$

拉格朗日函数为

$$L(\mathbf{w}_+,\mathbf{t},\boldsymbol{\xi},\boldsymbol{\alpha},\boldsymbol{\beta})=\frac{1}{2}\|\mathbf{t}\|_2^2+c\mathbf{1}^T\boldsymbol{\xi}+\frac{r}{2}\|\mathbf{w}_+\|_2^2+\boldsymbol{\alpha}^T(\mathbf{1}-\boldsymbol{\xi}+\mathbf{B}^T\mathbf{w}_+)-\boldsymbol{\beta}^T\boldsymbol{\xi}+\boldsymbol{\lambda}^T(\mathbf{t}-\mathbf{A}^T\mathbf{w}_+). \tag{13}$$

根据 KKT 条件，令 $\nabla_{\mathbf{w}_+}L=0,\nabla_{\mathbf{t}}L=0,\nabla_{\boldsymbol{\xi}}L=0$ 可以得到

$$\mathbf{w}_+=\frac{1}{r}(\mathbf{B}\boldsymbol{\alpha}-\mathbf{A}\boldsymbol{\lambda}), \quad \mathbf{t}=\boldsymbol{\lambda}, \quad c\mathbf{1}=\boldsymbol{\alpha}+\boldsymbol{\beta}. \tag{14}$$

[1] 这里指的是 $p=1$，即常规 SVM 的情况，对于 $p=2$，即采用平方铰链损失的 SVM，有 $\nabla_i f(\boldsymbol{\alpha})=\mathbf{w}^T\mathbf{x}_i+h-\alpha_i/(2c)$.

则可以得到对偶问题

$$\max_{\boldsymbol{\alpha},\boldsymbol{\lambda}} \frac{1}{2} \begin{bmatrix} \boldsymbol{\lambda}^T & \boldsymbol{\alpha}^T \end{bmatrix} \begin{bmatrix} \mathbf{A}^T\mathbf{A} + r\mathbf{I} & -\mathbf{A}^T\mathbf{B} \\ -\mathbf{B}^T\mathbf{A} & \mathbf{B}^T\mathbf{B} \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda} \\ \boldsymbol{\alpha} \end{bmatrix} - r \begin{bmatrix} \mathbf{0}^T & \mathbf{1}^T \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda} \\ \boldsymbol{\alpha} \end{bmatrix}, \quad \text{s.t. } 0 \leq \alpha_i \leq c. \tag{15}$$

其是规模为 n 的二次规划问题. 设其目标函数为 $f(\mathbf{s})$，其中

$$\mathbf{s} = \begin{bmatrix} \boldsymbol{\lambda} \\ \boldsymbol{\alpha} \end{bmatrix} \tag{16}$$

采用坐标下降法进行求解，首先注意到

$$\nabla_{\mathbf{s}} f = \left(\begin{bmatrix} r\mathbf{I} & \mathbf{O} \\ \mathbf{O} & \mathbf{O} \end{bmatrix} + \begin{bmatrix} \mathbf{A}^T\mathbf{A} & -\mathbf{A}^T\mathbf{B} \\ -\mathbf{B}^T\mathbf{A} & \mathbf{B}^T\mathbf{B} \end{bmatrix} \right) \begin{bmatrix} \boldsymbol{\lambda} \\ \boldsymbol{\alpha} \end{bmatrix} - r \begin{bmatrix} \mathbf{0} \\ \mathbf{1} \end{bmatrix} \tag{17}$$

$$= r \left(\begin{bmatrix} \boldsymbol{\lambda} \\ -\mathbf{1} \end{bmatrix} + \begin{bmatrix} -\mathbf{A}^T \\ \mathbf{B}^T \end{bmatrix} \mathbf{w}_+ \right) \tag{18}$$

根据前面坐标下降法的原理，可以将更新分为两个部分：

$\boldsymbol{\lambda}$ part: 更新 λ_i，由于 λ_i 是无约束的，因此有

$$\lambda_i := \lambda_i - \frac{\nabla_{\lambda_i} f(\mathbf{s})}{\|\mathbf{x}_i^+\|_2^2}. \tag{19}$$

此处 $\nabla_{\lambda_i} f(\mathbf{s}) = r(\lambda_i - \mathbf{w}_+^T \mathbf{x}_i^+)$. 并更新

$$\mathbf{w}_+ := \mathbf{w}_+ + \frac{\nabla_{\lambda_i} f(\mathbf{s})}{r\|\mathbf{x}_i^+\|_2^2} \mathbf{x}_i^+. \tag{20}$$

$\boldsymbol{\alpha}$ part: 根据前面分析，可以直接得到

$$\boldsymbol{\alpha}_i := \pi_{[0,c]} \left(\boldsymbol{\alpha}_i - \frac{\nabla_{\alpha_i} f(\mathbf{s})}{\|\mathbf{x}_i^-\|_2^2} \right). \tag{21}$$

而这里 $\nabla_{\alpha_i} f(\mathbf{s}) = r(-1 + \mathbf{w}_+^T \mathbf{x}_i^-)$. 并更新

$$\mathbf{w}_+ := \mathbf{w}_+ + \frac{1}{r}(\alpha_i^{new} - \alpha_i^{old}) \mathbf{x}_i^-. \tag{22}$$

1.2 逐次超松弛迭代法

逐次超松弛迭代 (Successive over-relaxation, SOR) 原本是用于解线性方程组 $\mathbf{Ax} = \mathbf{b}$ 的方法， [2] 将该方法拓展到 SVM 中二次规划问题的求解. 回顾原始的 QPP 问题

$$\min_{\boldsymbol{\alpha}} \frac{1}{2} \boldsymbol{\alpha}^T \mathbf{Q} \boldsymbol{\alpha} + \mathbf{h}^T \boldsymbol{\alpha}, \quad 0 \leq \alpha_i \leq c.$$

将矩阵 $\mathbf{Q}$ 加性分裂为 $\mathbf{Q} = \mathbf{L} + \mathbf{D} + \mathbf{L}^T$, 其中 $\mathbf{L}, \mathbf{D}$ 分别为严格下三角矩阵和对角矩阵. 则有以下的迭代公式

$$\boldsymbol{\alpha}^{(t+1)} = \pi_{[0,c]}(\boldsymbol{\alpha}^{(t)} - \omega \mathbf{D}^{-1}(\mathbf{h} + \mathbf{L}\boldsymbol{\alpha}^{(t+1)} + \mathbf{L}^T \boldsymbol{\alpha}^{(t)} + \mathbf{D}\boldsymbol{\alpha}^{(t)})), \quad \omega \in (0, 2). \tag{23}$$

或写成 [2]

$$\boldsymbol{\alpha}^{(t+1)} = \pi_{[0,c]}(\boldsymbol{\alpha}^{(t)} - \omega\mathbf{D}^{-1}(\nabla_{\boldsymbol{\alpha}} f(\boldsymbol{\alpha}) + \mathbf{L}(\boldsymbol{\alpha}^{(t+1)} - \boldsymbol{\alpha}^{(t)}))), \quad \omega \in (0,2). \tag{24}$$

以及逐元素更新的形式:

$$\alpha_i^{(t+1)} = \pi_{[0,c]}\left[\alpha_i^{(t)} - \frac{\omega}{Q_{ii}}\left(\sum_{j=1}^{i-1} Q_{ij}\alpha_j^{(t+1)} + \sum_{j=1}^{n} Q_{ij}\alpha_j^{(t)} + h_i\right)\right], \quad \omega \in (0,2). \tag{25}$$

我们称 ω 为松弛因子, 是手动设置的超参数. $\omega \in (0,2)$ 是 SOR 算法收敛的必要条件, 且 ω 取值往往会影响收敛速度, 理论上其最优的取值为

$$\omega^* = \frac{2}{1 + \sqrt{(1 - \rho^2(\mathbf{J}))}}. \tag{26}$$

其中 $\rho(\mathbf{J})$ 表示雅可比矩阵 $\mathbf{J} = -\mathbf{D}^{-1}(\mathbf{L} + \mathbf{L}^T)$ 的谱半径. 对于任意初值, SOR 算法均有 Q-线性收敛性, 即

$$f(\boldsymbol{\alpha}^{(t+1)}) - f(\boldsymbol{\alpha}^*) \leq r[f(\boldsymbol{\alpha}^{(t)}) - f(\boldsymbol{\alpha}^*)]. \tag{27}$$

SOR 算法的效率很高, 但缺点是需要存储至少 $n(n+1)/2$ 个元素, 因此其更适用于非超大规模的问题和稀疏矩阵的求解.

Example: TWSVM. 对于问题 (10), 可以写出其对应的对偶问题为

$$\min_{\boldsymbol{\alpha}} \frac{1}{2}\boldsymbol{\alpha}^T\mathbf{B}^T(\mathbf{A}^T\mathbf{A} + \mathbf{I})^{-1}\mathbf{B}\boldsymbol{\alpha} - \mathbf{1}^T\boldsymbol{\alpha}, \quad \text{s.t. } 0 \leq \alpha_i \leq c. \tag{28}$$

就可以采用 SOR 算法进行求解, 在实际实验中可以发现其求解速度很高.

参考文献

[1] C.-J. Hsieh, K.-W. Chang, C.-J. Lin, S. S. Keerthi, and S. Sundararajan, “A dual coordinate descent method for large-scale linear svm,” in *Proceedings of the 25th international conference on Machine learning*, pp. 408–415, 2008.

[2] O. L. Mangasarian and D. R. Musicant, “Successive overrelaxation for support vector machines,” *IEEE Transactions on Neural Networks*, vol. 10, no. 5, pp. 1032–1037, 1999.