对偶坐标下降法（DCDM）-L2R SVC

原问题

$$\min_{w} \quad \frac{1}{2}w^T w + C\sum_{i=1}^{l}\xi(w;x_i,,y_i) \tag{1}$$

其中$\xi(w;x_i,y_i)$是损失函数，C是惩罚项，L1损失和L2损失的损失函数如下；

$$max(1-y_i w^T x_i,0) \quad and \quad max(1-y_i w^T x_i,0)^2 \tag{2}$$

对偶问题

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \qquad & f(\boldsymbol{\alpha}) = \frac{1}{2}\boldsymbol{\alpha}^T\bar{Q}\boldsymbol{\alpha} - \boldsymbol{e}^T\boldsymbol{\alpha} \\ \text{subject to} \qquad & 0 \leq \alpha_i \leq U, \forall i, \end{aligned}$$

其中，$\bar{Q}=Q+D, D$是对角矩阵，并且$Q_{ij}=y_i y_j x_i^T x_j$.对于L1-SVC来讲$U=C, D_{ii}=0$，对于L2−SVC有$U=\infty, D_{ii}=1/2C$,

算法

DCDM共有两层循环，每一个外层循环k都要产生一个α^k向量；内层循环从 1 到 l （α的维度），一次只更新α的一个元素($\alpha^{k,i} \rightarrow \alpha^{k,i+1}$)，固定其他元素。一次内层循环求解一个单变量的子问题。

$$\boldsymbol{\alpha}^{k,i} = [\alpha_1^{k+1},\ldots,\alpha_{i-1}^{k+1},\alpha_i^k,\ldots,\alpha_l^k]^T,\ \forall i=2,\ldots,l.$$

k 表示第k次外层循环,i表示第i次内层循环。前 $i-1$个α的元素已经更新了之后，我们通过求解下面的单变量的子问题来实现$\alpha^{k,i} \rightarrow \alpha^{k,i+1}$的更新：

$$\min_{d} \ f(\boldsymbol{\alpha}^{k,i}+d\boldsymbol{e}_i) \quad \text{subject to} \ \ 0 \leq \alpha_i^k + d \leq U,$$

其中,

$$f(\boldsymbol{\alpha}^{k,i}+d\boldsymbol{e}_i) = \frac{1}{2}\bar{Q}_{ii}d^2 + \nabla_i f(\boldsymbol{\alpha}^{k,i})d + \text{constant},$$

$$\boldsymbol{e}_i = [0,\ldots,0,1,0,\ldots,0]^T$$

下面，我们对这个二次函数进行求解：

令$\Delta^P f(\alpha)$是投影梯度：

$$\nabla_i^P f(\boldsymbol{\alpha}) = \begin{cases} \nabla_i f(\boldsymbol{\alpha}) & \text{if } 0 < \alpha_i < U, \\ \min(0, \nabla_i f(\boldsymbol{\alpha})) & \text{if } \alpha_i = 0, \\ \max(0, \nabla_i f(\boldsymbol{\alpha})) & \text{if } \alpha_i = U. \end{cases}$$

容易得到，当$\Delta^P f(\alpha) = 0$的时候我们不需要更新$\alpha^{k,i}$,当$\bar{Q}_{ii} > 0$的时候，二次函数的极值点为$-\frac{\Delta_i f(\alpha^{k,i})}{\bar{Q}_{ii}}$,并且需要保证更新以后的$\alpha^{k,i+1}$在约束范围内，所以有：

$$\alpha_i^{k,i+1} = \min\left(\max\left(\alpha_i^{k,i} - \frac{\nabla_i f(\boldsymbol{\alpha}^{k,i})}{\bar{Q}_{ii}}, 0\right), U\right).$$

因为在更新$\alpha^{k,i+1}$的时候需要对$\Delta_i f(\alpha^{k,i})$进行求解：

$$\nabla_i f(\boldsymbol{\alpha}) = (\bar{Q}\boldsymbol{\alpha})_i - 1 = \sum\nolimits_{j=1}^{l} \bar{Q}_{ij}\alpha_j - 1.$$

因为$\bar{Q}_{ii}$可能很大，计算的时候比较复杂，所以在线性SVM里面，我们可以重新定义$\Delta_i f(\alpha^{k,i})$的求解过程：
因为：

$$\boldsymbol{w} = \sum\nolimits_{j=1}^{l} y_j \alpha_j \boldsymbol{x}_j,$$

所以我们有：

$$\nabla_i f(\boldsymbol{\alpha}) = y_i \boldsymbol{w}^T \boldsymbol{x}_i - 1 + D_{ii}\alpha_i.$$

参数w的更新规则：

$$\boldsymbol{w} \leftarrow \boldsymbol{w} + (\alpha_i - \bar{\alpha}_i) y_i \boldsymbol{x}_i.$$

并且我们可以初始化参数$\alpha^0 = 0, w = 0$,算法流程如下：

Algorithm 1 A dual coordinate descent method for Linear SVM

- Given $\boldsymbol{\alpha}$ and the corresponding $\boldsymbol{w} = \sum_i y_i \alpha_i \boldsymbol{x}_i$.
- While $\boldsymbol{\alpha}$ is not optimal

  For $i = 1, \ldots, l$

  (a) $G = y_i \boldsymbol{w}^T \boldsymbol{x}_i - 1 + D_{ii}\alpha_i$

  (b)

$$PG = \begin{cases} \min(G, 0) & \text{if } \alpha_i = 0, \\ \max(G, 0) & \text{if } \alpha_i = U, \\ G & \text{if } 0 < \alpha_i < U \end{cases}$$

  (c) If $|PG| \neq 0$,

  $\bar{\alpha}_i \leftarrow \alpha_i$

  $\alpha_i \leftarrow \min(\max(\alpha_i - G/\bar{Q}_{ii}, 0), U)$

  $\boldsymbol{w} \leftarrow \boldsymbol{w} + (\alpha_i - \bar{\alpha}_i) y_i \boldsymbol{x}_i$

子问题的随机扰动

在算法1中，我们对α的更新是按照顺序进行元素的更新的，即$\alpha_1, \alpha_2, \cdots, \alpha_l$,但是实验证明，以一个打乱的随机顺序解决子问题可能会加快收敛速度。所以我们打乱$\{1, \cdots l\} \rightarrow \{\pi(1), \cdots, \pi(l)\}$,即按照$\alpha_{\pi(1)}, \alpha_{\pi(2)}, \cdots, \alpha_{\pi(l)}$的顺序去更新参数。

$$\alpha_t^{k,i+1} = \alpha_t^{k,i} + \arg \min_{0 \le \alpha_t^{k,i} + d \le U} f(\boldsymbol{\alpha}^{k,i} + d\boldsymbol{e}_t) \text{ if } \pi_k^{-1}(t) = i.$$

Shrinking 策略

因为$0 \le \alpha_i \le U$,如果$\alpha = 0$或者$\alpha = U$的时候，那么α则可能保持不变。 为了加快求解，我们进行Shrinking去求解下面的子问题：

$$\begin{aligned} \min_{\alpha_A} \quad & \frac{1}{2}\alpha_A^T \bar{Q}_{AA}\alpha_A + (\bar{Q}_{AA}\alpha_A - e_A)^T \alpha_A \\ & s.t \quad 0 \le \alpha_i \le U, i \in A \end{aligned}$$

其中，A是对应的active α的集合，$\bar{A} = \{1, \cdots l\}/A$

shrinking 条件：

$$\alpha_i^{k,i} = 0 \text{ and } \nabla_i f(\boldsymbol{\alpha}^{k,i}) > \bar{M}^{k-1},$$
$$\alpha_i^{k,i} = U \text{ and } \nabla_i f(\boldsymbol{\alpha}^{k,i}) < \bar{m}^{k-1},$$

where

$$\bar{M}^{k-1} = \begin{cases} M^{k-1} & \text{if } M^{k-1} > 0, \\ \infty & \text{otherwise,} \end{cases}$$
$$\bar{m}^{k-1} = \begin{cases} m^{k-1} & \text{if } m^{k-1} < 0 \\ -\infty & \text{otherwise.} \end{cases}$$

$$M^{k-1} \equiv \max_j \nabla_j^P f(\boldsymbol{\alpha}^{k-1,j}), m^{k-1} \equiv \min_j \nabla_j^P f(\boldsymbol{\alpha}^{k-1,j}).$$

停止条件：

$$M^k - m^k < \epsilon$$

对偶坐标下降法（DCDM）-L2R LR

原问题

L2正则化逻辑回归目标是使以下函数最小化：

$$P^{\text{LR}}(\boldsymbol{w}) = C \sum_{i=1}^{l} \log \left(1 + e^{-y_i \boldsymbol{w}^T \boldsymbol{x}_i}\right) + \frac{1}{2} \boldsymbol{w}^T \boldsymbol{w},$$

对偶问题

$$\min_{\boldsymbol{\alpha}} \quad D^{\text{LR}}(\boldsymbol{\alpha}) = \frac{1}{2} \boldsymbol{\alpha}^T Q \boldsymbol{\alpha} + \sum_{i:\alpha_i > 0} \alpha_i \log \alpha_i + \sum_{i:\alpha_i < C} (C - \alpha_i) \log(C - \alpha_i)$$
$$\text{subject to} \quad 0 \le \alpha_i \le C, \ i = 1, \ldots, l,$$

其中$Q_{ij} = y_i y_j x_i^T x_j$

求解此问题的单变量子问题：

$$\begin{aligned} \min_{z} \quad & g(z) \equiv (c_1 + z)\log(c_1 + z) + (c_2 - z)\log(c_2 - z) + \frac{a}{2}z^2 + bz \\ \text{subject to} \quad & -c_1 \le z \le c_2, \end{aligned}$$

where

$$c_1 = \alpha_i, c_2 = C - \alpha_i, a = Q_{ii}, \text{ and } b = (Q\boldsymbol{\alpha})_i.$$

其中$\alpha^{k,i} \rightarrow \alpha^{k,i+1}$的更新为：$\alpha^{k,i+1} = \alpha^{k,i} + ze_i$

我们利用牛顿法对上述问题进行求解：

$$z^{k+1} = z^k + d, \quad d = -\frac{g'(z^k)}{g''(z^k)},$$

其中，

$$g'(z) = az + b + \log\frac{c_1 + z}{c_2 - z}, \text{ and } \quad g''(z) = a + \frac{c_1 + c_2}{(c_1 + z)(c_2 - z)}.$$

同样的，为了加速计算，我们重写$(Q\alpha)_i$:
因为:

$$\boldsymbol{w}(\boldsymbol{\alpha}) \equiv \sum_{i=1}^{l} y_i \alpha_i \boldsymbol{x}_i.$$

所以我们有：

$$(Q\boldsymbol{\alpha})_i = \sum_{j=1}^{l} y_i y_j \alpha_j \boldsymbol{x}_j^T \boldsymbol{x}_i = y_i (\sum_{j=1}^{l} y_j \alpha_j \boldsymbol{x}_j^T) \boldsymbol{x}_i = y_i \boldsymbol{w}(\boldsymbol{\alpha})^T \boldsymbol{x}_i.$$

参数w的更新规则；

$$\boldsymbol{w}(\boldsymbol{\alpha} + z\boldsymbol{e}_i) = \boldsymbol{w}(\boldsymbol{\alpha}) + z y_i \boldsymbol{x}_i,$$