Liblinear 中的多分类模型

倪杰

2021 年 4 月 22 日

摘要

libsvm 中处理多分类问题还是采用拆分的方法。把多分类问题转化为二分类问题。liblinear 中提供了一种直接在模型中采用多分类思想的 svm 模型。其 motivation 来自于纠错编码。

目录

1 Motivation

$S = \{(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)\}$
$Y = \{1, 2, \ldots, k\}$ 其中 k 为类的个数
考虑 l 个二分类器，每个分类器在训练时将 Y 中两个不相交互补子集标记为 $\{1, -1\}$
把这 l 个分类器及其在每个类上的标记写成编码矩阵 M。
每一列代表一个分类器，每一行代表一个类的编码.

$h_1(x)$	$h_2(x)$		$h_l(x)$
+1	+1	......	+1
+1	+1	......	-1
+1	-1	......	-1
-1	-1	......	-1
......	......	......	......
-1	-1	......	-1

k

l

对于样本 x, 分类结果可写为 $h(x) = (h_1(x), h_2(x), \ldots, h_l(x))$, 为一串编码。
记 M_r 为矩阵 M 的第 r 行，即第 r 类的编码
最终分类结果可写为 $H(x) = arg\max_{\mathbf{r}} h(x) \cdot M_r$, 即编码最相近的类
这种编码矩阵的方式可扩展到实数矩阵，其中每一个数代表权重或者置信。判断准则仍是内积最大原则.
第 i 个样本的损失函数可写为:

$$\max_r \{h(x_i) \cdot M_r + 1 - \delta_{y_i,r}\} - h(x_i) \cdot M_{y_i}$$

对于分类正确的样本 x_i, 有 $\forall i, r \quad h(x_i) \cdot M_{y_i} - h(x_i) \cdot M_r \geq 1 - \delta_{y_i,r}$
引入软间隔，有:
$\forall i, r \quad h(x_i) \cdot M_{y_i} - h(x_i) \cdot M_r \geq 1 - \delta_{y_i,r} - \xi_i$
整个优化问题写为:

$$\begin{aligned} \min_M \quad & ||M||_p + C \sum_{i=1}^{l} \xi_i && (1) \\ subject\ to: \quad & \forall i, r \quad h(x_i) \cdot M_{y_i} - h(x_i) \cdot M_r \geq 1 - \delta_{y_i,r} - \xi_i && (2) \\ & && (3) \end{aligned}$$

在 SVM 情景下，令 $h(x) = x$, 采用 Frobenius 范数，就可得 liblinear 中的优化问题:

$$
\begin{aligned}
\min_{\boldsymbol{w}_m,\xi_i} \quad & \frac{1}{2}\sum_{m=1}^{k}\boldsymbol{w}_m^T\boldsymbol{w}_m + C\sum_{i=1}^{l}\xi_i \\
\text{subject to} \quad & \boldsymbol{w}_{y_i}^T\boldsymbol{x}_i - \boldsymbol{w}_m^T\boldsymbol{x}_i \geq e_i^m - \xi_i,\ \ i = 1,\dots,l,
\end{aligned}
$$

$$
e_i^m = \begin{cases} 0 & \text{if } y_i = m, \\ 1 & \text{if } y_i \neq m. \end{cases}
$$

分类结果为 $\arg\max_{m=1,\dots,k}\omega_m^T x$

对偶问题如下:

$$
\begin{aligned}
\min_{\boldsymbol{\alpha}} \quad & \frac{1}{2}\sum_{m=1}^{k}\|\boldsymbol{w}_m\|^2 + \sum_{i=1}^{l}\sum_{m=1}^{k} e_i^m\alpha_i^m \\
\text{subject to} \quad & \sum_{m=1}^{k}\alpha_i^m = 0, \forall i = 1,\dots,l \\
& \alpha_i^m \leq C_{y_i}^m, \forall i = 1,\dots,l, m = 1,\dots,k,
\end{aligned}
$$

$$
\boldsymbol{w}_m = \sum_{i=1}^{l}\alpha_i^m\boldsymbol{x}_i, \forall m, \quad \boldsymbol{\alpha} = [\alpha_1^1,\dots,\alpha_1^k,\dots,\alpha_l^1,\dots,\alpha_l^k]^T.
$$

$$
C_{y_i}^m = \begin{cases} 0 & \text{if } y_i \neq m, \\ C & \text{if } y_i = m. \end{cases}
$$

2 sequential dual method

对偶问题如果算上平凡的约束，一共有 kl 个变量，直接求解并不方便，考虑将该优化问题拆分为关于 i 的子优化问题，对这些子优化问题之间采用坐标下降法。

$$
\begin{aligned}
\min_{\bar{\boldsymbol{\alpha}}_i} \quad & \sum_{m=1}^{k} \frac{1}{2} A (\alpha_i^m)^2 + B_m \alpha_i^m \\
\text{subject to} \quad & \sum_{m=1}^{k} \alpha_i^m = 0, \\
& \alpha_i^m \leq C_{y_i}^m, m = \{1, \ldots, k\},
\end{aligned}
$$

where

$$
A = \boldsymbol{x}_i^T \boldsymbol{x}_i \text{ and } B_m = \boldsymbol{w}_m^T \boldsymbol{x}_i + e_i^m - A \alpha_i^m. \tag{18}
$$

In (18), A and B_m are constants obtained using $\boldsymbol{\alpha}$ of the previous iteration..

子问题之间的坐标下降法如下图算法所示:

Algorithm 1 The coordinate descent method for (15)

- Given $\boldsymbol{\alpha}$ and the corresponding $\boldsymbol{w}_m$
- While $\boldsymbol{\alpha}$ is not optimal, (outer iteration)
  1. Randomly permute $\{1, \ldots, l\}$ to $\{\pi(1), \ldots, \pi(l)\}$
  2. For $i = \pi(1), \ldots, \pi(l)$, (inner iteration)

     If $\bar{\boldsymbol{\alpha}}_i$ is active and $\boldsymbol{x}_i^T \boldsymbol{x}_i \neq 0$ (i.e., $A \neq 0$)
     - Solve a $|U_i|$-variable sub-problem (19)
     - Maintain $\boldsymbol{w}_m$ for all m by (20)

其中更新准则 (20) 为:

$\boldsymbol{w}_m \leftarrow \boldsymbol{w}_m + (\alpha_i^m - \hat{\alpha}_i^m) y_i \boldsymbol{x}_i.$

3 solving the sub problem

对偶子问题经过 shirinking 之后为:

$$\begin{aligned} \min_{\bar{\boldsymbol{\alpha}}_i^{U_i}} \quad & \sum_{m \in U_i} \frac{1}{2} A(\alpha_i^m)^2 + B_m \alpha_i^m \\ \text{subject to} \quad & \sum_{m \in U_i} \alpha_i^m = -\sum_{m \notin U_i} \alpha_i^m, \\ & \alpha_i^m \le C_{y_i}^m, m \in U_i. \end{aligned} \tag{19}$$

由 KKT 条件，可得:

$$\begin{aligned} \sum_{m \in U_i} \alpha_i^m &= -\sum_{m \notin U_i} C_{y_i}^m, \\ \alpha_i^m &\le C_{y_i}^m, \forall m \in U_i, \\ \rho_m(C_{y_i}^m - \alpha_i^m) &= 0, \rho_m \ge 0, \forall m \in U_i, \\ A\alpha_i^m + B_m - \beta &= -\rho_m, \forall m \in U_i. \end{aligned}$$

Using (22), equations (23) and (24) are equivalent to

$$\begin{aligned} & A\alpha_i^m + B_m - \beta = 0, \text{ if } \alpha_i^m < C_{y_i}^m, \forall m \in U_i, \\ & A\alpha_i^m + B_m - \beta = AC_{y_i}^m + B_m - \beta \le 0, \text{ if } \alpha_i^m = C_{y_i}^m, \forall m \in U_i. \end{aligned}$$

定义 $D_m = B_m + C_{y_i}^m$

如果 β 已知，可以看出 $\alpha_i^m = \min\{C_{y_i}^m, \frac{\beta - B_m}{A}\}$

$$\beta = \frac{\sum_{m \in U_i, \alpha_i^m < C_{y_i}^m} D_m - AC}{|\{m \mid m \in U_i, \alpha_i^m < C_{y_i}^m\}|}.$$

β 的求解可以用一种 sequential 的方法求解，可以证明这样找出的 β 就是符合条件的 β

$$h = \frac{\sum_{m \in \Phi} D_m - AC}{|\Phi|} \ge \max_{m \notin \Phi} D_m.$$

整体算法如下图所示:

Algorithm 2 Solving the sub-problem

- Given A, $\boldsymbol{B}$
- Compute $\boldsymbol{D}$ by (27)
- Sort $\boldsymbol{D}$ in decreasing order; assume $\boldsymbol{D}$ has elements $D_1, D_2, \ldots, D_{|U_i|}$
- $r \leftarrow 2, \beta \leftarrow D_1 - AC$
- While $r \leq |U_i|$ and $\beta/(r-1) < D_r$
  1. $\beta \leftarrow \beta + D_r$
  2. $r \leftarrow r + 1$
- $\beta \leftarrow \beta/(r-1)$
- $\alpha_i^m \leftarrow \min(C_{y_i}^m, (\beta - B_m)/A), \forall m$

4 stopping condition

E.4 Stopping Condition

The KKT optimality conditions of (15) imply that there are $b_1, \ldots, b_l \in R$ such that for all $i = 1, \ldots, l$, $m = 1, \ldots, k$,

$$\begin{aligned}\boldsymbol{w}_m^T \boldsymbol{x}_i + e_i^m - b_i = 0 \text{ if } \alpha_i^m < C_i^m,\\ \boldsymbol{w}_m^T \boldsymbol{x}_i + e_i^m - b_i \leq 0 \text{ if } \alpha_i^m = C_i^m.\end{aligned}$$

Let

$$G_i^m = \frac{\partial f(\boldsymbol{\alpha})}{\partial \alpha_i^m} = \boldsymbol{w}_m^T \boldsymbol{x}_i + e_i^m, \forall i, m,$$

the optimality of $\boldsymbol{\alpha}$ holds if and only if

$$\max_m G_i^m - \min_{m:\alpha_i^m < C_i^m} G_i^m = 0, \forall i. \tag{34}$$

At each inner iteration, we first compute G_i^m and define:

$$\text{minG} \equiv \min_{m:\alpha_i^m < C_i^m} G_i^m, \text{maxG} \equiv \max_m G_i^m, S_i = \text{maxG} - \text{minG}.$$

Then the stopping condition for a tolerance $\epsilon > 0$ can be checked by

$$\max_i S_i < \epsilon. \tag{35}$$

Note that maxG and minG are calculated based on the latest $\boldsymbol{\alpha}$ (i.e., $\boldsymbol{\alpha}$ after each inner iteration).