Incremental Semi-supervised Support Vector Clustering

Huang Xiao[1], Bojan Kolosnjaji

No Institute Given

Abstract. Keywords: Incremental learning, semi-supervised clustering, novelty detection

1 One-class SVM

Here is an introduction of how one-class SVM works. It is based on the work by [**?**]. Let's consider the training data,

$$x_1, \ldots, x_l \in \mathcal{X} \tag{1}$$

where l is the number of training samples. We pick up Kernel function defined as,

$$k(x, y) = \langle \Phi(x) \cdot \Phi(y) \rangle \tag{2}$$

For instance, the widely used Gaussian kernel,

$$k(x, y) = e^{-\lambda \|x-y\|^2} \tag{3}$$

Now the functional $f(x)$ defines a hyperplane separating input data $\mathcal{X}$ into two classes, where most of samples are classified as $+1$, a minority of data are classified as -1. To solve this problem, we have following quadratic problem:

$$\underset{w \in F, \boldsymbol{\xi} \in \mathbb{R}^l, \rho \in \mathbb{R}}{\arg\min} \; \frac{1}{2}\|w\|^2 + \frac{1}{vl}\sum_i \xi_i - \rho \tag{4}$$

$$\text{subject to} \;\; (w \cdot \Phi(x_i)) \geq \rho - \xi_i, \;\; \xi_i \geq 0 \tag{5}$$

The decision function is based on

$$y(x) = sgn\left(\sum_i \alpha_i k(x_i, x) - \rho\right) \tag{6}$$

The problem is solved by Lagrangian multipliers, and converted to Dual. We also note that for each point x_i with $0 < \alpha_i < 1/(vl)$, it lies exactly on the hyperplane such that we can get ρ by any of these samples (Support Vectors).

$$\rho = \sum_j \alpha_j k(x_j, x_i) \tag{7}$$

However, solving this dual problem with the box constraint given is numerically not stable, while the dual variables $\boldsymbol{\alpha}$ are in fact very small. Thus, in LibSVM[**?**] we solve the scaled dual problem as follows,

$$\begin{aligned} \mathcal{W} = & \quad \min \alpha^T K \alpha \\ \text{subject to} & \quad \textstyle\sum_i \alpha_i = vl \\ & 0 \leq \alpha_i \leq 1, \ \ i = 1, \cdots, l \end{aligned} \tag{8}$$

Therefore, in LibSVM implementation both the dual variables $\{\alpha\}$ and the intercept ρ are scaled such that the decision function is also scaled by vl.

2 Incremental One-Class SVMs

Here we introduce the incremental version of OC-SVMs based on incremental SVMs [**?**]. Gradient of $\mathcal{W}$ w.r.t. α_i is,

$$g_i = \frac{\partial \mathcal{W}}{\partial \alpha_i} = \sum_j \alpha_j Q_{i,j} = f(x_i) + \rho$$

where $f(x_i) = w \cdot \Phi(x_i) - \rho$ and $Q_{i,j} = k(x_i, x_j)$. We also denote a new function

$$h_i = g_i - \rho = \begin{cases} > 0 & \alpha_i = 0 \text{ and } x_i \in \mathcal{R} \\ = 0 & 0 < \alpha_i < 1 \text{ and } x_i \in \mathcal{S} \\ < 0 & \alpha_i = 1 \text{ and } x_i \in \mathcal{E} \end{cases}$$

where $\mathcal{S}, \mathcal{R}, \mathcal{E}$ denote set of support vectors, reserved vectors, and error vectors. Adding a new point x_c into the training set $\mathcal{X}$, we have initially set $\alpha_c = 0$,

$$\Delta h_i = (Q_{ic} - Q_{sc}) \Delta\alpha_c + \sum_{j \in \mathcal{S}} (Q_{ij} - Q_{sj}) \Delta\alpha_j, \ \ x_i \in \mathcal{X} \cup x_c \tag{9}$$

$$\Delta\alpha_c + \sum_{j \in \mathcal{S}} \Delta\alpha_j = v \tag{10}$$

Note that x_s is any support vector in $\mathcal{S}$, for convenience, we simply use x_{s_1}. For $x_i \in \mathcal{S}$, we have $h_i \equiv 0$ such that we can construct a linear equation system,

$$\mathcal{Q} \cdot \begin{bmatrix} v \\ \Delta\alpha_{s_1} \\ \Delta\alpha_{s_2} \\ \vdots \\ \Delta\alpha_{s_v} \end{bmatrix} = - \begin{bmatrix} 1 \\ Q_{s_2 c} - Q_{s_1 c} \\ \vdots \\ Q_{s_v c} - Q_{s_1 c} \end{bmatrix} \Delta\alpha_c \tag{11}$$

where we have $(s_1, \cdots, s_v)$ are the indices for $\mathcal{S}$ and $\|\mathcal{S}\| = v$, and the $\mathcal{Q}$ is $v \times (v+1)$ matrix,

$$\mathcal{Q} = \begin{bmatrix} -1 & 1 & \cdots & 1 \\ 0 & Q_{s_2 s_1} - Q_{s_1 s_1} & \cdots & Q_{s_2 s_v} - Q_{s_1 s_v} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & Q_{s_v s_1} - Q_{s_1 s_1} & \cdots & Q_{s_v s_v} - Q_{s_1 s_v} \end{bmatrix}$$

3 Reweighed Regularization

Given a set of supervised labels $\mathcal{D}_M = \{x_i, y_i\}_{i=1}^M$, where $y_i = \{+1, -1\}$, we compute the impact scores of relevant samples with respect to their kernel similarity to the given labeled samples,

$$f(x_i) = \frac{1}{M_i^+} \sum_{j=1}^{|M_i^+|} k(x_i, x_j) - \frac{1}{M_i^-} \sum_{l=1}^{|M_i^-|} k(x_i, x_l)$$

Moreover, we restrict the sets M_i^+ and M_i^- as being set $D_M^i = \{x | k(x, x_i) \leq h\}$

4 Support Vector Clustering

Here we introduce another formulation of one-class SVM firstly presented in Benhur's work [**?**]. Alternatively we can search for a compact enough hypersphere in the feature space such that the ball enclose most of the training samples while permitting a small portion of data (outliers) lying outside the hypersphere. This hypersphere is defined by the radius R and a center $\boldsymbol{a}$, then the objective of this problem is to find a smallest hyperball to enclose all the training samples. Similar as SVM, a soft margin formulation is adopted by introducing slack variables $\{\xi_i\}_{i=1}^N$, where $\xi_i \geq 0$, we have the objective function,

$$\begin{aligned} &\min_R R^2 + C \sum_i \xi_i \\ \text{s.t. } &\|\Phi(x_i) - \boldsymbol{a}\|^2 \leq R^2 + \xi_i \\ &\xi_i \geq 0, \ \forall i \in \{1, \ldots, N\} \end{aligned} \tag{12}$$

Similar as described in [**?**], we leverage Lagrangian multipliers to solve this problem, such that we get,

$$L = R^2 - \sum_j \left(R^2 + \xi_j - \|\Phi(x_j) - \mathbf{a}\|^2\right) \alpha_j - \sum_j \xi_j \mu_j + \sum_j C\xi_j, \tag{13}$$

Moreover, we derive optimal conditions as follows,

$$\sum \alpha_j = 1, \tag{14}$$

$$\mathbf{a} = \sum \alpha_j \Phi(x_j), \tag{15}$$

$$\alpha_j = C - \mu_j \tag{16}$$

According to KKT conditions, we also have,

$$\xi_j \mu_j = 0, \tag{17}$$

$$\left(R^2 + \xi_j - \|\Phi(x_j) - \mathbf{a}\|^2\right) \alpha_j = 0 \tag{18}$$

Substitute Eq.(14)-(18) back into the Lagrangian, we obtain the Wolfra dual form as,

$$\mathcal{W} = \sum_{i,j} \alpha_i \alpha_j k(x_i, x_j) - \sum_j k(x_j, x_j)\alpha_j, \tag{19}$$

$$\text{s.t.} \quad 0 \leq \alpha_j \leq C, \ j = 1, \ldots, N. \tag{20}$$

where $k(x_i, x_j) = \langle \Phi(x_i), \Phi(x_j) \rangle$ denote the kernel function. To solve this quadratic programing problem with box constraints, we will employ the SMO-type optimizer introduced in the following section.

5 SMO Optimization

In this section, we employed the SMO-type optimization technique to solve the dual problem, originally presented in [**?**]. However, we apply the working set selection algorithm using second order information described in Fan *et al.*, [**?**].

The SMO-type optimization proposed to solve the QP problem in an iterative way, where only two variables are considered at each iteration. Due to the linear constraint in Eq.(20), this is the minimal number of variables permitted at each iteration, such that the objective function is guaranteed to reduced at every round. For simplicity, we denote α_1 and α_2 as the first and the second variables, whereas we know the following conditions hold given the rest variables unchanged,

$$\alpha_1 + \alpha_2 = 1 - \sum_{i=3}^{N} \alpha_i = \varDelta \tag{21}$$

$$L = \max(0, \varDelta - C) \ \text{ and } \ H = \min(C, \varDelta) \tag{22}$$

Here we explicitly express the lower bound L and upper bound H for α_2. The objective can be now formulated as function of α_1 and α_2.

$$\begin{aligned} \mathcal{W} = {} & \alpha_1^2 k_{11} + \alpha_2^2 k_{22} + 2\alpha_1\alpha_2 k_{12} + 2\alpha_1 V_1 + 2\alpha_2 V_2 \\ & -\alpha_1 k_{11} - \alpha_2 k_{22} + \underbrace{\sum_{i,j=3}^{N} \alpha_i \alpha_j k_{ij} - \sum_{j=3}^{N} \alpha_j k_{jj}}_{\text{constant}} \end{aligned} \tag{23}$$

where $V_i = \sum_{j=3}^{N} \alpha_j k_{ij}$ and we replace α_1 with $\varDelta - \alpha_2$ and take the first derivative, we can get the extremum when the derivative is 0, that is,

$$\alpha_2 = \alpha_2^* + \frac{f(x_2) - f(x_1)}{2\eta} \tag{24}$$

where $\eta = k_{11} + k_{22} - 2k_{12}$ is the second derivative of objective $\mathcal{W}$ and $f(x_i)$ is the squared distance of x_i to the center. Therefore, we get the update equation of α_2

given the α_1, which will maximally reduce the objective function. However, this is the updating rule when the second derivative is strictly positive, if $\eta <= 0$, this implies the extremum of the objective function occurs at the bounds, therefore if $\eta <= 0$ we compute the objective functions at both bounds and pick the lower value as the new α_2.

$$\begin{aligned} W_L - W_H =& (\mathit{\Delta}^2 - \mathit{\Delta})(k_{11} - k_{22}) + 2\mathit{\Delta}(V_1 - V_2), \text{ if } L = 0, H = \mathit{\Delta} && (25) \\ W_L - W_H =& (\mathit{\Delta}^2 - 2C\mathit{\Delta} + \mathit{\Delta} - 2C)k_{11} - (\mathit{\Delta}^2 - 2C\mathit{\Delta} - \mathit{\Delta} + 2C)k_{22} && (26) \\ & - (4C - 2\mathit{\Delta})\mathit{\Delta}(V_1 - V_2), \text{ if } L = \mathit{\Delta} - C, H = C \end{aligned}$$

Now the question is to find the maximal violating pair x_i and x_j such that optimizing the objective function with respect to α_i and α_j can approximately reduce the objective function. We follow the working set selection strategy **WSS3** in Fan *et al.*.[**?**]. At each iteration, we update the squared ball center $\|\mathbf{a}\|^2$ and the gradients $\frac{\partial \mathcal{W}}{\partial \boldsymbol{\alpha}}$ denoted as $\mathit{\Delta} f(\boldsymbol{\alpha})$.

$$\begin{aligned} \|\mathbf{a}\|^2 =& \|\mathbf{a}^*\|^2 + (\alpha_1^2 - \alpha_1^{*2})k_{11} + (\alpha_2^2 - \alpha_2^{*2})k_{22} + && (27) \\ & (2\alpha_1\alpha_2 - 2\alpha_1^*\alpha_2^*)k_{12} + (\alpha_{\mathbf{B}} - \alpha_{\mathbf{B}}^*)^T K_{\mathbf{BN}} \boldsymbol{\alpha}_{\mathbf{N}} \\ \mathit{\Delta} f(\boldsymbol{\alpha}) =& \mathit{\Delta} f(\boldsymbol{\alpha})^* + 2K_{\mathbf{NB}}(\alpha_{\mathbf{B}} - \alpha_{\mathbf{B}}^*) && (28) \end{aligned}$$