Approximate Leave-One-Out with kernel SVM

Wenda Zhou, Peng Xu

March 13, 2019

1 C-Support Vector Classification

Let $\boldsymbol{K}$ denote a positive-definite kernel matrix (hence invertible), with $\boldsymbol{K}_{i,j} = K(\boldsymbol{x}_i, \boldsymbol{x}_j)$. The objective of a kernel SVC can be expressed in the "loss + penalty" variational form (notice the optimal $\boldsymbol{\gamma}^*$ satisfies $\boldsymbol{\gamma}^* = \boldsymbol{y} \odot \boldsymbol{\alpha}$, with $\boldsymbol{\alpha}$ being the dual solution):

$$\min_{\rho,\gamma} \sum_{j=1}^{n} \max\left[0, 1 - y_j h(x_j)\right] + \frac{\lambda}{2}\boldsymbol{\gamma}^\top \boldsymbol{K}\boldsymbol{\gamma}, \qquad h(x_j) = \boldsymbol{K}_{\cdot,j}^\top \gamma + \rho. \tag{1}$$

For simplicity we ignore the offset ρ for now. Let S and V be the smooth set and the set of singularities, respectively. For the j-th observation, $j \in S$, we have

$$\dot{\ell}(\boldsymbol{K}_{\cdot,j}^\top \gamma) = -y_j \cdot \mathbf{1}\{y_j \boldsymbol{K}_{\cdot,j}^\top \gamma < 1\}, \qquad \ddot{\ell}(\boldsymbol{K}_{\cdot,j}^\top \gamma) = 0.$$

Additionally,

$$\nabla R(\boldsymbol{\gamma}) = \lambda \boldsymbol{K}\boldsymbol{\gamma}, \qquad \nabla^2 R(\boldsymbol{\gamma}) = \lambda \boldsymbol{K}.$$

Substitute corresponding terms in Thm. 4.1, we deduce the ALO formula for kernel SVC:

$$\boldsymbol{K}_{\cdot,i}^\top \tilde{\gamma}^{\setminus i} = \boldsymbol{K}_{\cdot,i}^\top \hat{\gamma} + a_i g_{\ell,i},$$

where

$$a_i = \begin{cases} \dfrac{1}{\lambda}\boldsymbol{K}_{\cdot,i}^\top \left[\boldsymbol{K}^{-1} - \boldsymbol{K}^{-1}\boldsymbol{K}_{\cdot,V}\left(\boldsymbol{K}_{\cdot,V}^\top \boldsymbol{K}^{-1}\boldsymbol{K}_{\cdot,V}\right)^{-1}\boldsymbol{K}_{\cdot,V}^\top \boldsymbol{K}^{-1}\right]\boldsymbol{K}_{\cdot,i} & i \in S, \\ \left[\lambda\left(\boldsymbol{K}_{\cdot,V}^\top \boldsymbol{K}^{-1}\boldsymbol{K}_{\cdot,V}\right)_{ii}^{-1}\right]^{-1} & i \in V, \end{cases}$$

and

$$g_{\ell,S} = -y_S \odot \mathbf{1}\left\{y_S \boldsymbol{K}_{\cdot,S}^\top \gamma < 1\right\}, \qquad g_{\ell,V} = \left(\boldsymbol{K}_{\cdot,V}^\top \boldsymbol{K}_{\cdot,V}\right)^{-1}\boldsymbol{K}_{\cdot,V}^\top \left[\sum_{j \in S: y_j \boldsymbol{K}_{\cdot,j}^\top \gamma < 1} y_j \boldsymbol{K}_{\cdot,j} - \lambda \boldsymbol{K}\alpha\right].$$

2 ε-Support Vector Regression

The ε-SVR is associated with the ε-insensitive loss function. Its objective function can be written as (notice the optimal $\boldsymbol{\gamma}^*$ satisfies $\boldsymbol{\gamma}^* = \boldsymbol{\alpha}^* - \boldsymbol{\alpha}$, where $\boldsymbol{\alpha}$ and $\boldsymbol{\alpha}^*$ are the dual solutions):

$$\min_{\rho,\gamma} \sum_{j=1}^{n} \max\left[0, |y_j - h(x_j)| - \varepsilon\right] + \frac{\lambda}{2}\gamma^\top \boldsymbol{K}\boldsymbol{\gamma}, \qquad h(x_j) = \boldsymbol{K}_{\cdot,j}^\top \gamma + \rho. \tag{2}$$

So for the j-th observation, $j \in S$, we have

$$\dot{\ell}(\boldsymbol{K}_{\cdot,j}^\top \gamma) = -\operatorname{sgn}\left(\boldsymbol{K}_{\cdot,j}^\top \gamma\right) \cdot \mathbf{1}\left\{\left|y_j - \boldsymbol{K}_{\cdot,j}^\top \gamma\right| \geq \varepsilon\right\}, \qquad \ddot{\ell}(\boldsymbol{K}_{\cdot,j}^\top \gamma) = 0.$$

Our ALO recipe will then be exactly the same as in C-SVC, except

$$g_{\ell,S} = -\operatorname{sgn}\left(\boldsymbol{K}_{\cdot,S}^\top \gamma\right) \odot \mathbf{1}\left\{\left|y_j - \boldsymbol{K}_{\cdot,S}^\top \gamma\right| \geq \varepsilon\right\},$$

and

$$g_{\ell,V} = \left(\boldsymbol{K}_{\cdot,V}^\top \boldsymbol{K}_{\cdot,V}\right)^{-1} \boldsymbol{K}_{\cdot,V}^\top \left[\sum_{j \in S: \left|y_j - \boldsymbol{K}_{\cdot,j}^\top \gamma\right| \geq \varepsilon} \operatorname{sgn}\left(\boldsymbol{K}_{\cdot,j}^\top \gamma\right) \boldsymbol{K}_{\cdot,j} - \lambda \boldsymbol{K} \alpha\right].$$

3 ν-SVC and ν-SVR

ν-SVC and ν-SVR are introduced in [Sch+00], with $\nu \in (0, 1]$ a new parameter that can control the number of support vectors. Notably, ν replaces C in C-SVC and ε in ε-SVR, and in the latter case ε itself becomes an optimization argument.

Because of the following equivalencies, there is no need to figure out the variational problem in order to derive ALO for ν-SVC and ν-SVR:

- Let α^* be the dual solution of ν-SVC with parameters (C, ν), then α^*/ρ^* is an optimal solution of C-SVC with $C = 1/(n\rho^*)$. See [CL01].
- Let α^* be the dual solution of ν-SVR with parameters (C, ν), then α^* is also the solution of ε-SVR with parameters $(C/n, \varepsilon^*)$. See [CL02].

Here ρ^* and ε^* are the corresponding primal solutions, and can be worked out quickly from α^* through equations derived from KKT conditions. See [CL11, Section 4.2].

4 Multiclass-classifier

To handle the multinomial classification, LIBSVM adopts the one-versus-one strategy: for a dataset with $y \in \mathcal{S}$, $|\mathcal{S}| = K$, LIBSVM will train $K(K-1)/2$ classifiers $\hat{f}_{(u,v)}(\boldsymbol{x})$, one for each 2-combination (u, v) of $\mathcal{S}$. The prediction for a new observation $\boldsymbol{x}_{\text{new}}$ is then obtained through voting: $\hat{y}_{\text{new}} = \text{mode}\{\hat{f}_{(u,v)}(\boldsymbol{x}_{\text{new}})\}$, and the label with the lowest index will be chosen in case of ties.

Note that for a specific pair (u', v'), the leave-i-out prediction $f_{(u',v')}^{\backslash i}(\boldsymbol{x}_j)$ is exactly the full-data prediction $f_{(u',v')}(\boldsymbol{x}_j)$ when y_i is neither u' nor v', for $\boldsymbol{x}_i$ does not contribute to the training of $f_{(u',v')}(\cdot)$ in anyway. Therefore, to obtain the multiclass ALO prediction $\tilde{y}_j^{\backslash i}$, we first construct all $K(K-1)/2$ full-data predictions, then replace the $K-1$ ones whose training are affected by y_i with the corresponding binary ALO predictions, and finally ensemble the result through popular voting.

5 Computation

To compute ALO for SVM, we must perform a variety of operations with the kernel matrix $\boldsymbol{K} \in \mathbb{R}^{n \times n}$. In practice n can often be quite large, therefore, naïvely implement the ALO formula will result in severe loss of efficiency. This issue can be, however, alleviated by exploiting the symmetry of $\boldsymbol{K}$: consider the Cholesky decomposition $\boldsymbol{K} = \boldsymbol{L}\boldsymbol{L}^\top$, then for an arbitrary index set $E \subseteq \{1, \ldots, n\}$, we have

$$\boldsymbol{L}^{-1}\boldsymbol{K}_{\cdot,E} = \boldsymbol{L}^{-1}\boldsymbol{L}\boldsymbol{L}_{E,\cdot}^\top = \boldsymbol{L}_{E,\cdot}^\top.$$

By further considering the thin QR factorization $\boldsymbol{L}_{V,\cdot}^\top = \boldsymbol{Q}_V\boldsymbol{R}_V$, where $\boldsymbol{Q}_V$ semi-orthogonal and $\boldsymbol{R}_V$ upper-triangular, we may expressed a_s, $s \in S$, as a difference between two quadratic forms:

$$\begin{aligned}
\lambda a_s &= \boldsymbol{K}_{,s}^\top\boldsymbol{K}^{-1}\boldsymbol{K}_{,s} - \boldsymbol{K}_{,s}^\top\boldsymbol{K}^{-1}\boldsymbol{K}_{\cdot,V}\left(\boldsymbol{K}_{\cdot,V}^\top\boldsymbol{K}^{-1}\boldsymbol{K}_{\cdot,V}\right)^{-1}\boldsymbol{K}_{\cdot,V}^\top\boldsymbol{K}^{-1}\boldsymbol{K}_{,s} \\
&= \boldsymbol{K}_{,s}^\top(\boldsymbol{L}\boldsymbol{L}^\top)^{-1}\boldsymbol{K}_{,s} - \boldsymbol{K}_{,s}^\top(\boldsymbol{L}\boldsymbol{L}^\top)^{-1}\boldsymbol{K}_{\cdot,V}\left(\boldsymbol{K}_{\cdot,V}^\top(\boldsymbol{L}\boldsymbol{L}^\top)^{-1}\boldsymbol{K}_{\cdot,V}\right)^{-1}\boldsymbol{K}_{\cdot,V}^\top(\boldsymbol{L}\boldsymbol{L}^\top)^{-1}\boldsymbol{K}_{,s} \\
&= (\boldsymbol{L}^{-1}\boldsymbol{K}_{,s})^\top\boldsymbol{L}^{-1}\boldsymbol{K}_{,s} - (\boldsymbol{L}^{-1}\boldsymbol{K}_{,s})^\top(\boldsymbol{L}^{-1}\boldsymbol{K}_{\cdot,V})\left[(\boldsymbol{L}^{-1}\boldsymbol{K}_{\cdot,V})^\top(\boldsymbol{L}^{-1}\boldsymbol{K}_{\cdot,V})\right]^{-1}(\boldsymbol{L}^{-1}\boldsymbol{K}_{\cdot,V})^\top(\boldsymbol{L}^{-1}\boldsymbol{K}_{,s}) \\
&= \boldsymbol{L}_{s,\cdot}\boldsymbol{L}_{s,\cdot}^\top - \boldsymbol{L}_{s,\cdot}\boldsymbol{L}_{V,\cdot}^\top\left(\boldsymbol{L}_{V,\cdot}\boldsymbol{L}_{V,\cdot}^\top\right)^{-1}\boldsymbol{L}_{V,\cdot}\boldsymbol{L}_{s,\cdot}^\top \\
&= \boldsymbol{L}_{s,\cdot}\boldsymbol{L}_{s,\cdot}^\top - (\boldsymbol{Q}_V^\top\boldsymbol{L}_{s,\cdot})^\top(\boldsymbol{Q}_V^\top\boldsymbol{L}_{s,\cdot}).
\end{aligned}$$

Similarly, a_v with $v \in V$ can be evaluated through the inner product of two vectors:

$$(\lambda a_v)^{-1} = \left(\boldsymbol{L}_{V,\cdot}\boldsymbol{L}_{V,\cdot}^\top\right)_{v,v}^{-1} = \left(\boldsymbol{R}_V^\top\boldsymbol{Q}_V^\top\boldsymbol{Q}_V\boldsymbol{R}_V\right)_{v,v}^{-1} = \boldsymbol{R}_{v,\cdot}^{-1}\boldsymbol{R}_{\cdot,v}^{-1}.$$

As a result, with $\boldsymbol{L}$, and thus $\boldsymbol{L}_{S,\cdot}$, $\boldsymbol{L}_{V,\cdot}$ computed beforehand, we avoided direct multiplication of large matrices and reduced computational costs.

References

[CL01] Chih-Chung Chang and Chih-Jen Lin. "Training v-Support Vector Classifiers: Theory and Algorithms". In: *Neural Computation* 13.9 (2001), pp. 2119–2147. DOI: 10.1162/089976601750399335.

[CL02] Chih-Chung Chang and Chih-Jen Lin. "Training v-Support Vector Regression: Theory and Algorithms". In: *Neural Computation* 14.8 (2002), pp. 1959–1977. DOI: 10.1162/089976602760128081.

[CL11] Chih-Chung Chang and Chih-Jen Lin. “LIBSVM: A Library for Support Vector Machines”. In: *ACM Trans. Intell. Syst. Technol.* 2.3 (May 2011), 27:1–27:27. ISSN: 2157-6904. DOI: 10.1145/1961189.1961199. URL: https://www.csie.ntu.edu.tw/~cjlin/papers/libsvm.pdf.

[Sch+00] Bernhard Schölkopf et al. “New Support Vector Algorithms”. In: *Neural Computation* 12.5 (2000), pp. 1207–1245. DOI: 10.1162/089976600300015565.