Approximate Leave-One-Out with kernel SVM

Peng Xu

February 14, 2019

1 ALO with the Variational Problems

1.1 C-Support Vector Classification

Let $\boldsymbol{K}$ denote the positive-definite kernel matrix (hence invertible), with $\boldsymbol{K}_{i,j} = K(\boldsymbol{x}_i, \boldsymbol{x}_j)$. By the Representer Theorem, the dual problem to kernel SVC can be expressed in "loss + penalty" form:

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

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

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

Additionally,

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

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

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

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

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

1.2 ε-Support Vector Regression

For kernel SVR, the objective is

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

For the j-th observation, $j \in S$, we now have

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

Thus, our recipe will be exactly the same as in SVC except now

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

and

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

1.3 ν-Support Vector Classification

This section is inaccurate for now.

For ν-SVC, $\nu \in (0, 1]$

$$\min_{\rho,\boldsymbol{\alpha},b} \frac{1}{n\nu\rho} \sum_{j=1}^{n} \max\left[0, \rho - y_j f(x_j)\right] + \frac{1}{2\nu\rho}\boldsymbol{\alpha}^\top \boldsymbol{K}\boldsymbol{\alpha}, \qquad f(x_j) = \boldsymbol{K}_{\cdot,j}^\top \boldsymbol{\alpha} + b. \tag{3}$$

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

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

Additionally,

$$\nabla R(\boldsymbol{\alpha}) = \frac{1}{\nu\rho}\boldsymbol{K}\boldsymbol{\alpha}, \qquad \nabla^2 R(\boldsymbol{\alpha}) = \frac{1}{\nu\rho}\boldsymbol{K}.$$

The ALO formula for ν-SVC is then:

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

where

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

and

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

An issue with this implementation is that ρ will not be returned when solving the dual problem. Let T_+ and T_- be the index sets of identical size s that correspond to $y_i = \pm 1$ s.t. $0 < \alpha_i < 1$, respectively. Then one way to recover ρ is (Schölkopf *et al.*, 2000):

$$\rho = \frac{1}{2s} \left[\sum_{i \in T_+} \sum_{j=1}^{n} \alpha_j y_j \boldsymbol{K}_{i,j} - \sum_{i \in T_-} \sum_{j=1}^{n} \alpha_j y_j \boldsymbol{K}_{i,j} \right].$$

2 ALO with Approximate Explicit Feature Maps

In non-linear SVM, kernel trick is employed to avoid the explicit computation of feature maps, which sometime is impossible since the feature space can be infinite-dimensional. However, when sample size n is large, the kernel matrices become quite expensive to handle. Methods such as the Nyström approximation are used in order to retain the benefit of features mapping whilst retaining the speed of linear SVM. We may adopt a similar idea to help the ALO computation.

Let $\boldsymbol{X}$ be the data matrix and $\boldsymbol{K}$ be the corresponding kernel matrix. An approximation $\hat{\boldsymbol{\Phi}}$ to the feature maps $\boldsymbol{\Phi}(\boldsymbol{X})$ can be constructed as following (procedure adopted from `scikit-learn`):

1. Perform SVD: $\boldsymbol{K} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^{\top}$;
2. Clamp the singular values: $\tilde{\boldsymbol{S}} = \max(\boldsymbol{S}, 10^{-12})$;
3. Construct the approximate map as $\hat{\boldsymbol{\Phi}} = \boldsymbol{K}\boldsymbol{U}\tilde{\boldsymbol{S}}^{-1/2}\boldsymbol{V}^{\top} \approx \boldsymbol{K}^{1/2}$.

To compute ALO, we then simply replace the data matrix $\boldsymbol{X}$ with $\hat{\boldsymbol{\Phi}}$ in the linear SVM formula.

3 Numerical Experiment

We tested the two methods using simulated data. All model are fitted on the following grid of penalty: $\lambda = \exp(-2 : 6 : (1/3))$.

3.1 RBF

Both methods produce similar and satisfying result. Note that for $p > n$ example the scale of Y-axis is relatively small, so the classification error is not really as bad as it looks.

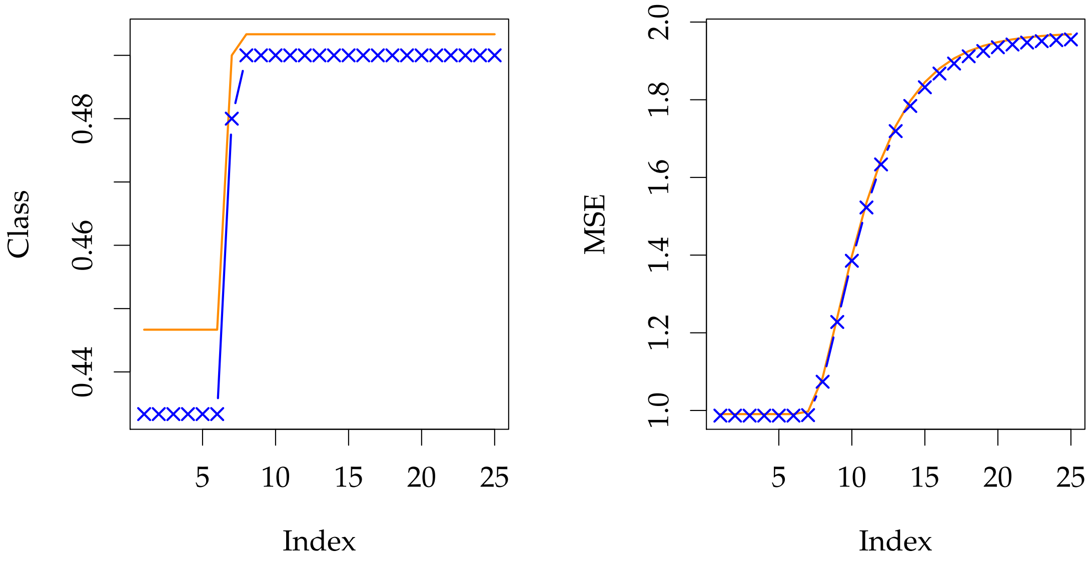

Figure 1: ALO and LOO comparison for SVM with RBF kernel. Direct method. $n = 300$, $p = 400$, $\gamma = 3/p$.

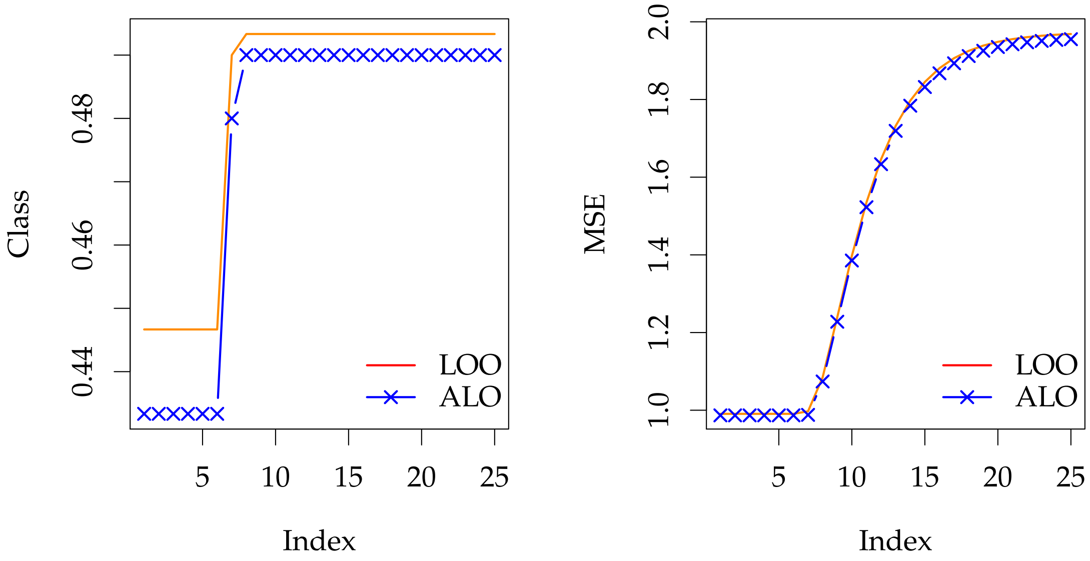

Figure 2: ALO and LOO comparison for SVM with RBF kernel. Feature map method. $n = 300$, $p = 400$, $\gamma = 3/p$.

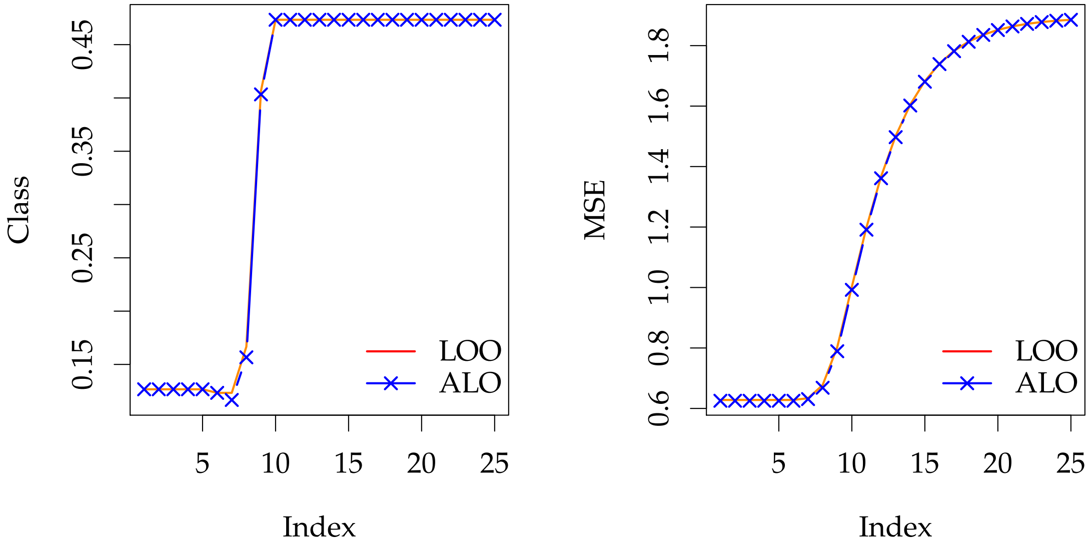

Figure 3: ALO and LOO comparison for SVM with RBF kernel. Direct method. $n = 300$, $p = 50$, $\gamma = 2/p$.

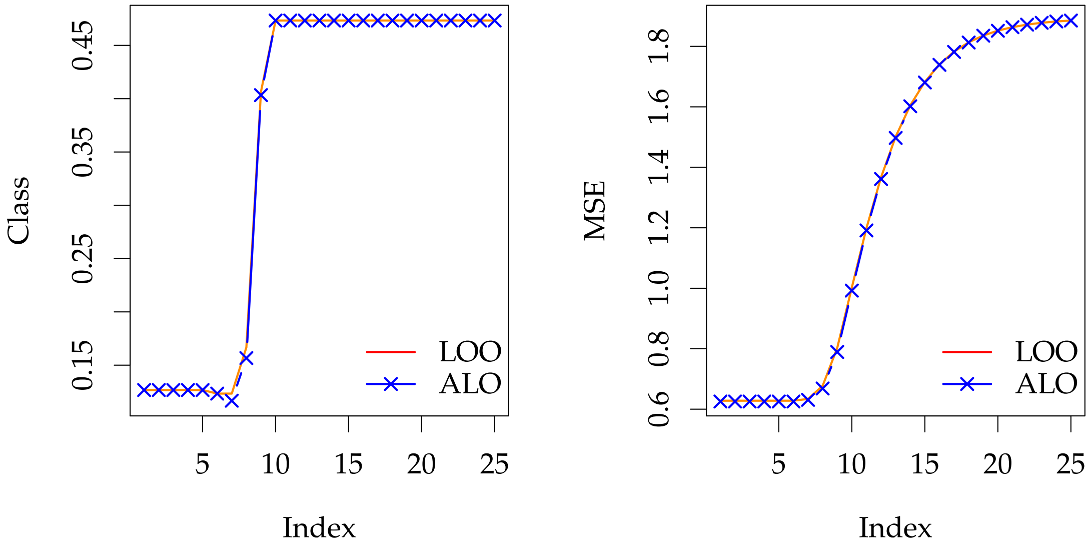

Figure 4: ALO and LOO comparison for SVM with RBF kernel. Feature map method. $n = 300$, $p = 50$, $\gamma = 2/p$.

3.2 Polynomial

Both methods again produce similar and satisfying result.

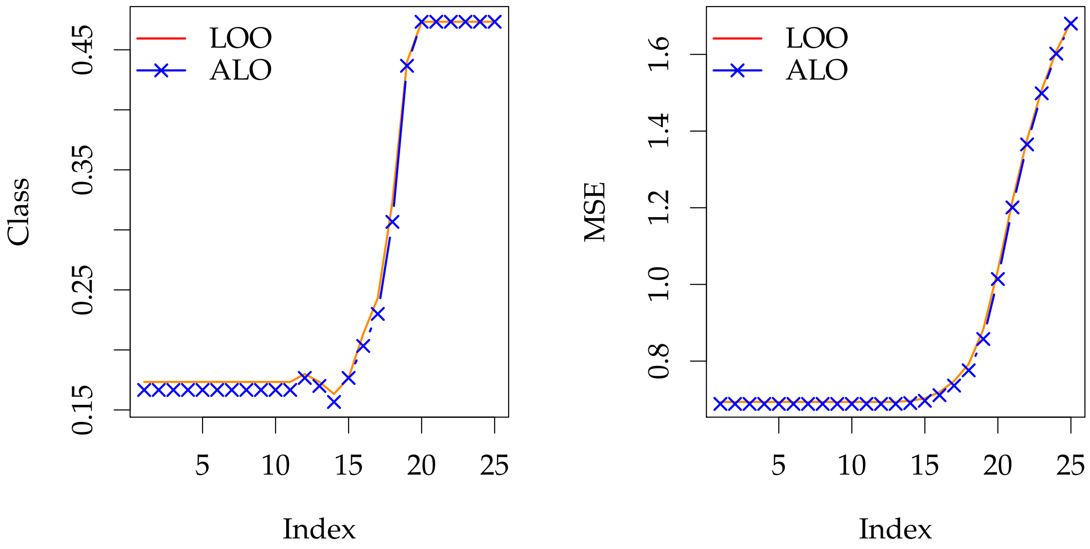

Figure 5: ALO and LOO comparison for SVM with polynomial kernel. Direct method. $n = 300$, $p = 50$, $\gamma = 3/p$, $d = 3$.

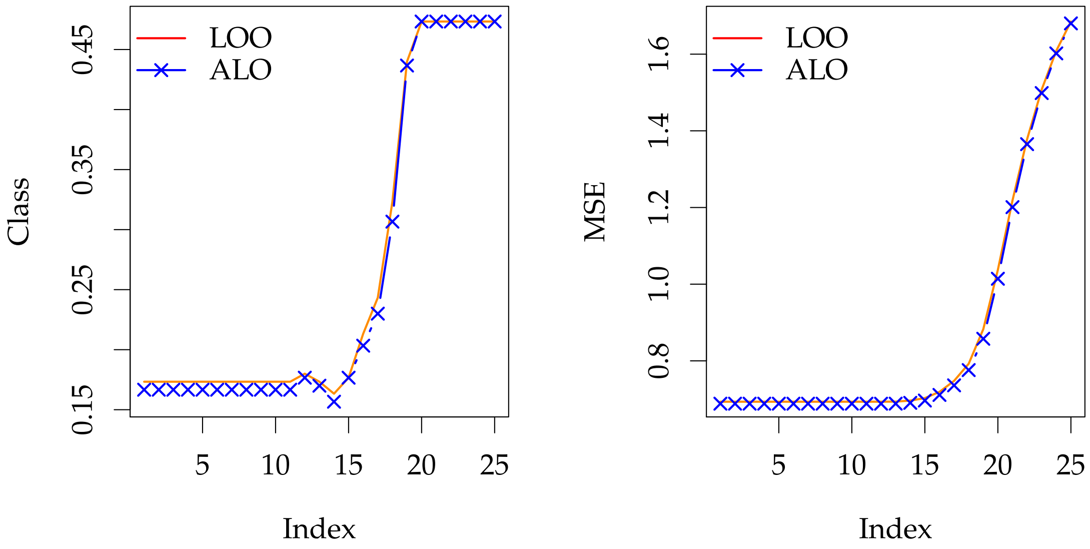

Figure 6: ALO and LOO comparison for SVM with RBF kernel. Feature map method. $n = 300$, $p = 50$, $\gamma = 3/p$, $d = 3$.

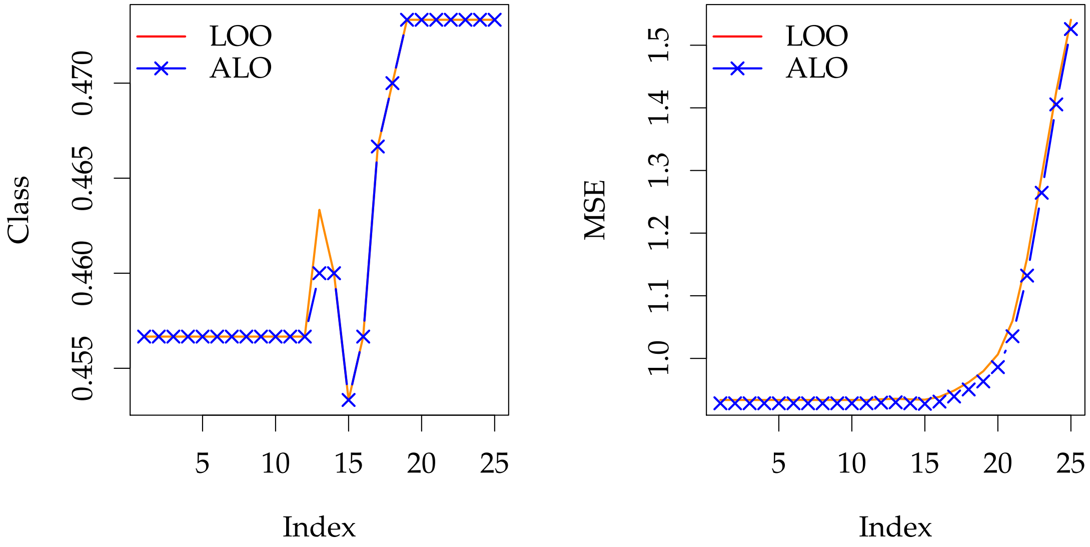

Figure 7: ALO and LOO comparison for SVM with polynomial kernel. Direct method. $n = 300$, $p = 50$, $\gamma = 2/p$, $c_0 = 0.25$, $d = 5$.

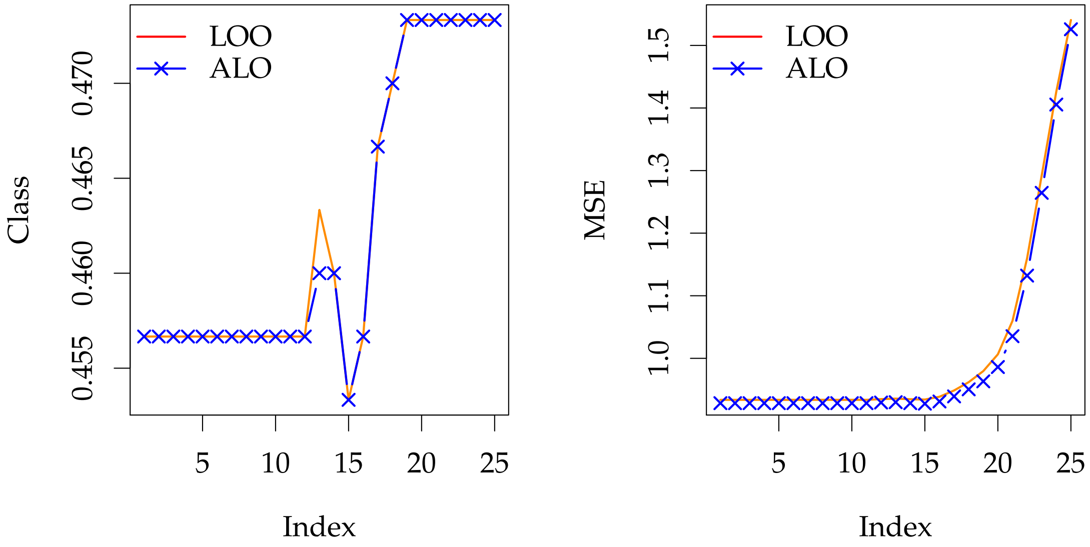

Figure 8: ALO and LOO comparison for SVM with RBF kernel. Feature map method. $n = 300$, $p = 50$, $\gamma = 2/p$, $c_0 = 0.25$, $d = 5$.

3.3 Sigmoid

None of the methods works well with the sigmoid kernel, however. Note that it is not a positive-definite kernel.

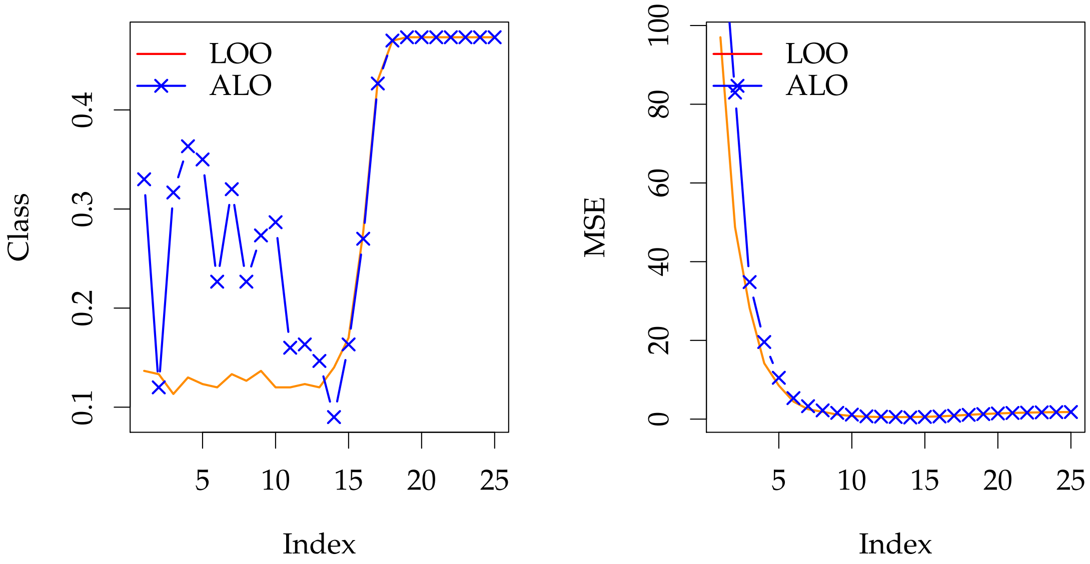

Figure 9: ALO and LOO comparison for SVM with sigmoid kernel. Direct method. $n = 300$, $p = 50$, $\gamma = 2/p$, $c_0 = 0.3$.

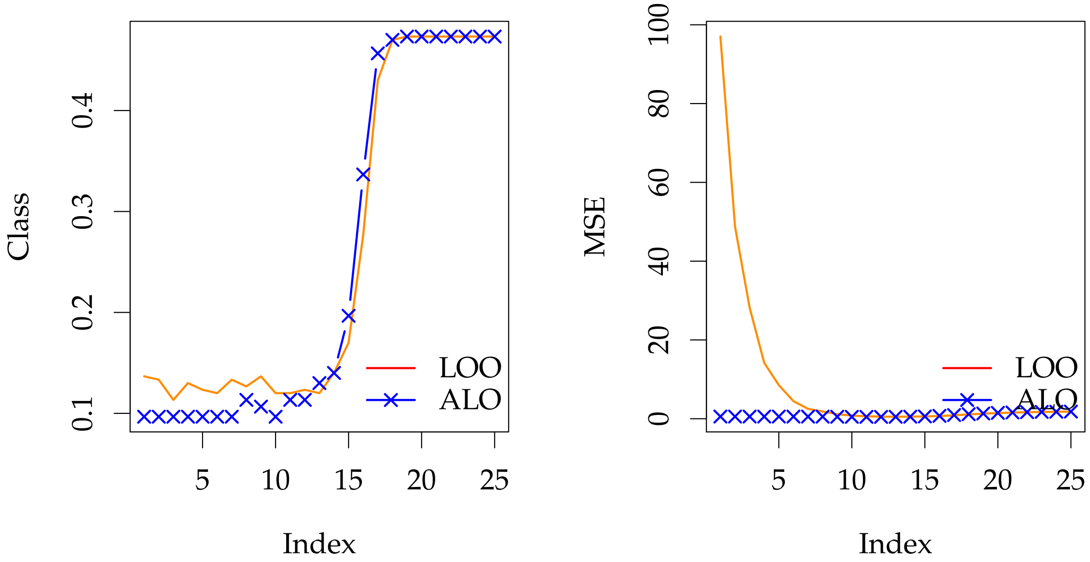

Figure 10: ALO and LOO comparison for SVM with sigmoid kernel. Feature map method. $n = 300$, $p = 50$, $\gamma = 2/p$, $c_0 = 0.3$.