SMM conditional probability and kernel

Chanwoo Lee, April 09, 2020

1 Strong duality

Our primal problem for SMM is

$$\begin{aligned} &\min_{U,V,\boldsymbol{\xi}} \frac{1}{2}\|UV^T\|^2 + C\sum_{i=1}^{N}\xi_i \\ &\text{subject to } y_i\left(\langle UV^T, X_i\rangle + b\right) \geq 1-\xi_i \\ &\xi_i \geq 0, \quad i=1,\ldots,N. \end{aligned} \tag{1}$$

If we consider U and V at the same time, there is no guarantee that the strong duality holds. However, if we consider U fixing V and vice versa, we have the strong duality for each sub iteration. Suppose that we fix V to update U, then (5) becomes

$$\begin{aligned} &\min_{U} \frac{1}{2}\operatorname{Vec}(V)^T \operatorname{diag}(\underbrace{V^TV,\cdots,V^TV}_{\text{total m}})\operatorname{Vec}(V) + C\boldsymbol{\xi}^T\mathbb{1} \\ &\text{subject to } \boldsymbol{\xi} + Z^T\operatorname{Vec}(U) + b\boldsymbol{y} \geq 1, \\ &\qquad\qquad \boldsymbol{\xi} \geq 0, \\ &\text{where } Z = (\operatorname{Vec}(y_1X_1V),\cdots,\operatorname{Vec}(y_NX_nV))^T \end{aligned} \tag{2}$$

Since Equation (2) is a Quadratic programming problem with linear constraints, we have strong duality for the alternating update algorithm.

2 SVM conditional probability calculation

In simulation, I made inseparable data set generating $(x_1,y_1),\ldots,(x_{30},y_{30})$ where $x_i \in \mathbb{R}^2$ and $y \in \{-1,1\}$. I randomly assigned y as 1 or -1 and add $(1.5,1.5)^T$ to feature vectors x if $y=1$. Figure 1 shows the generated data. First, I calculated conditional probability $\mathbb{P}(y=1|x)$ with the linear kernel. Figure 3 shows that the weird phenomenon we talked before still happens in the right bottom part. In nonlinear SVM case with radial kernel, classification boundary changes as in Figure 3 according to π values. Based on the weighted hinge loss classification, I calculate the conditional probability and Figure 4 shows the result.

3 SMM conditional probability calculation

To calculate conditional probability, we solve the regularization problem based on weighted hinge loss.

$$\begin{aligned} &\min_{U,V,\boldsymbol{\xi}} \frac{1}{2}\|UV^T\|^2 + C\left[(1-\pi)\sum_{y_i=1}\xi_i + \pi\sum_{y_i=-1}\xi_i\right] \\ &\text{subject to } y_i\left(\langle UV^T, X_i\rangle + b\right) \geq 1-\xi_i \end{aligned} \tag{3}$$

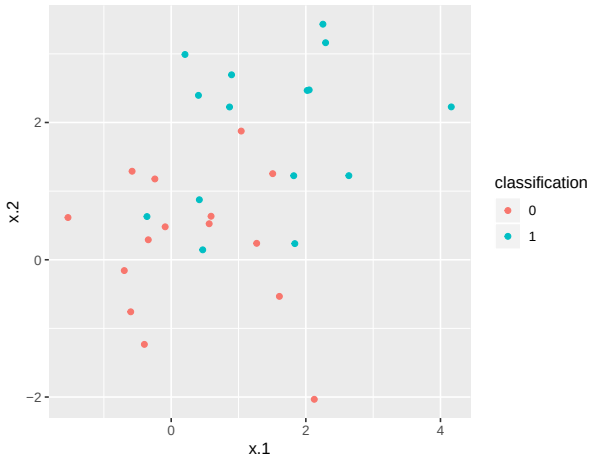

Figure 1: Simulation data points. Red colored points has $y = 1$ and blue colored points has $y = -1$.

$$\xi_i \geq 0, \quad i = 1, \ldots, N.$$

To solve this problem, we take the alternating update approach. We update U fixing V with the following dual problem.

$$\begin{aligned} \max_{\boldsymbol{\alpha}} \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \langle X_i H_V, X_j H_V \rangle, \\ \text{subject to } 0 \leq \alpha_i \leq C(1-\pi) \text{ for } y_i = 1, \\ 0 \leq \alpha_i \leq C\pi \text{ for } y_i = -1, \\ \sum_{i=1}^{N} \alpha_i y_i = 0. \end{aligned}$$

From this (3), we have updated $U = \sum_{i=1}^{N} \alpha_i y_i X_i V (V^T V)^{-1}$. We can update V fixing U by the same way. If we have theoretical guarantee that

$$\text{sign}(\langle X, UV^T \rangle + b) \to \text{sign}(\mathbb{P}(y = 1|X) - \pi),$$

we can obtain the conditional probability. My guess is that if the targeted rank is large enough, we have the result. To build this guess, consider the regularization problem of minimizing

$$\frac{1}{N} \sum_{i=1}^{N} L(y_i)[1 - y_i f(X_i))]_+ \frac{1}{C} \|h\|^2_{H_k}, \tag{4}$$

over all the functions of the form $f(X) = h(X) + b$, with $h \in H_k$. In our situation, $L(1) = 1 - \pi$ and $L(-1) = \pi$. There are two components: a data fit functional component and a regularization penalty component. The data fit component usually approaches a limiting functional as $N \to \infty$. In general, the target function is the minimizer of the limiting functional. Under the assumption that the target function can be approximated by the elements in the Reproducing Kernel Hilbert Space (RKHS) with other general regularity conditions, the solution of (4) approaches the target function as $N \to \infty$ [Lin, 2002]. The following lemma suggests that the solution of (4) approaches $\text{sign}[\mathbb{P}(Y = 1|X) - \frac{L(Y=-1)}{L(-1)+L(1)}]$[Lin et al., 2002]. In this sense, as long as considered reproducing

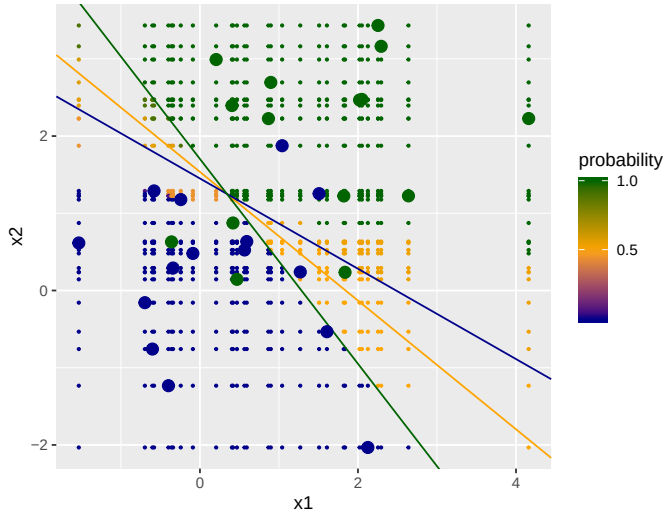

Figure 2: Green large points are from given data with $y = 1$. Blue large points are from $y = -1$. The yellow line is classification boundary when $\pi = 0.5$. The blue line and green line are when $\pi = 0.01$ and $\pi = 0.99$ respectively. Small dots are calculated conditional probability from the grid.

kernel Hilbert space is rich enough to approximate the target function, we have consistent estimator. This explains better performance in estimating thke conditional probability in nonlinear case (you can compare Figure and Figure).

Lemma 1. *The minimizer of* $E[L(Y)(1 - Yf(X))_+]$ *is sign*$[\mathbb{P}(Y = 1|X) - \frac{L(Y=-1)}{L(-1)+L(1)}]$.

4 Kernel SMM method

The suggested kernel method is to use $K(X, X') = g(\langle H_U X H_V, H_U X' H_V \rangle)$. In SVM case, we are looking for good hyper plane that separate the feature well including the kernel method (in kernel, hyper plane is in the extended dimension space). To make it clear, let us consider kernel method in SVM. For nonlinear SVM, we set the basis function $h : \mathbb{R}^{\dim(\boldsymbol{x})} \to \mathbb{R}^d$ where $d > \dim(\boldsymbol{x})$ is the usual case. Once the basis function are selected. the procedure is the same as before. We fit the SV classifier using input feature $h(\boldsymbol{x}_i) = (h_1(\boldsymbol{x}_i), \ldots, h_d(\boldsymbol{x}_i)), i = 1, \ldots, N$ and the produce the nonlinear function $\hat{f}(\boldsymbol{x}) = \langle h(\boldsymbol{x}), \hat{\boldsymbol{w}} \rangle + \hat{b} = \sum_{i=1}^{N} \hat{\alpha}_i y_i K(\boldsymbol{x}, \boldsymbol{x}_i) + \hat{b}$, where $K(\boldsymbol{x}, \boldsymbol{x}') = \langle h(\boldsymbol{x}), h(\boldsymbol{x}') \rangle$. Therefore, even in nonlinear case, we are finding the hyper plane in extended dimension. However, if we use $K(X, X') = g(\langle H_U X H_V, H_U X' H_V \rangle)$, we are obtaining classification function as $f(X) = \sum_{i=1}^{N} \alpha_i y_i g(\langle H_U X_i H_V, H_U X H_V \rangle) + b$. From the SVM case, we can interpret

$$g(\langle H_U X H_V, H_U X' H_V \rangle) = \langle H_{U'} h(X) H_{V'}, H_{U'} h(X') H_{V'} \rangle.$$

This interpretation is possible because as long as gram matrix K is positive semi definite and symmetric, we can find induced U', V', and $h : \mathbb{R}^{m \times n} \to \mathbb{R}^{m' \times n'}$ from K (need to verify). From this h, our primal SMM problem becomes,

$$\min_{U', V', \boldsymbol{\xi}} \frac{1}{2} \|U'(V')^T\|^2 + C \sum_{i=1}^{N} \xi_i \tag{5}$$
$$\text{subject to } y_i \left(\langle h(X_i), U'(V')^T \rangle + b \right) \geq 1 - \xi_i$$

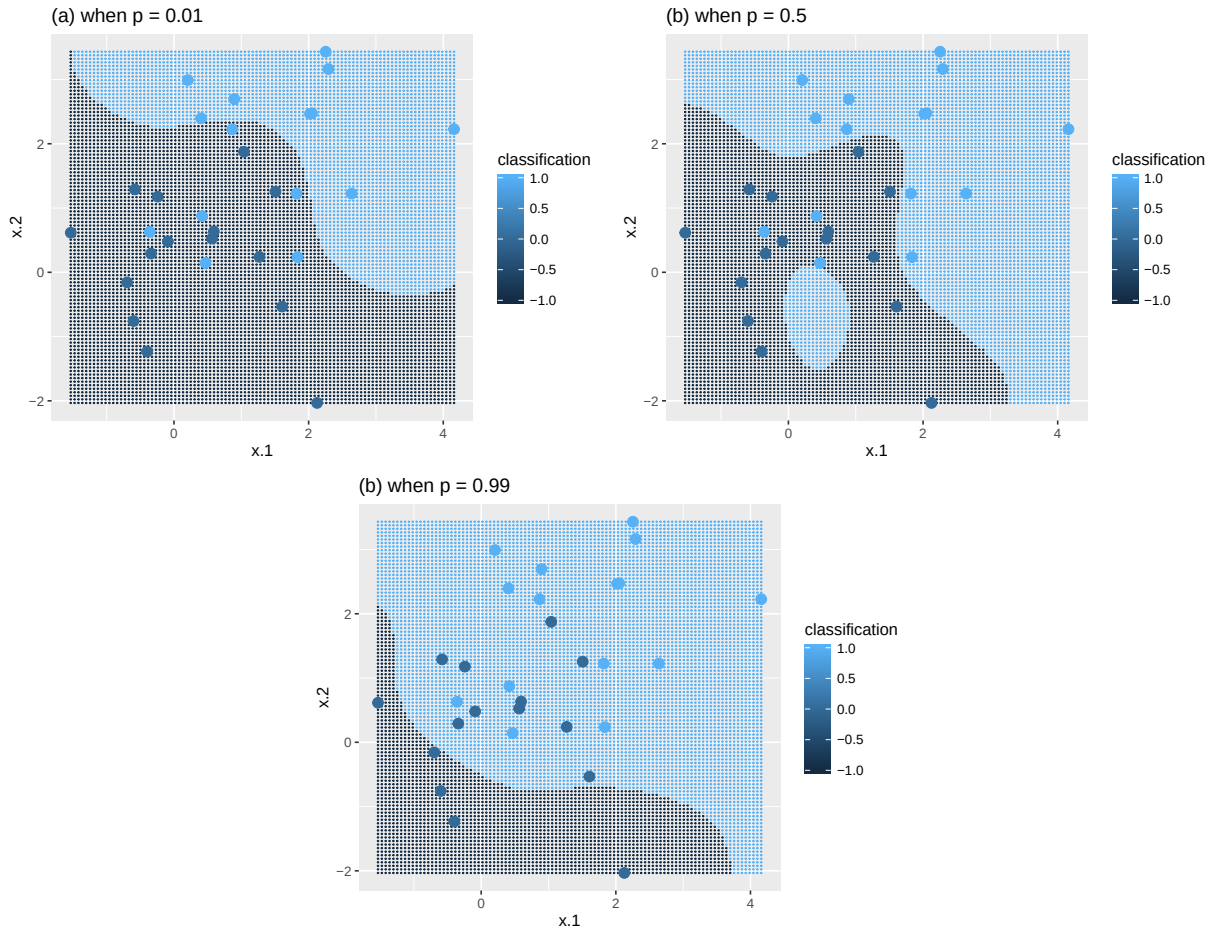

Figure 3: The figures show weighted hinge loss SVM classifier with radial kernel when $\pi \in \{0.01, 0.5, 0.99\}$. Black dots are classified as -1 and blue dots as 1. Sub figure (b) is the regular SVM with radial kerenl.

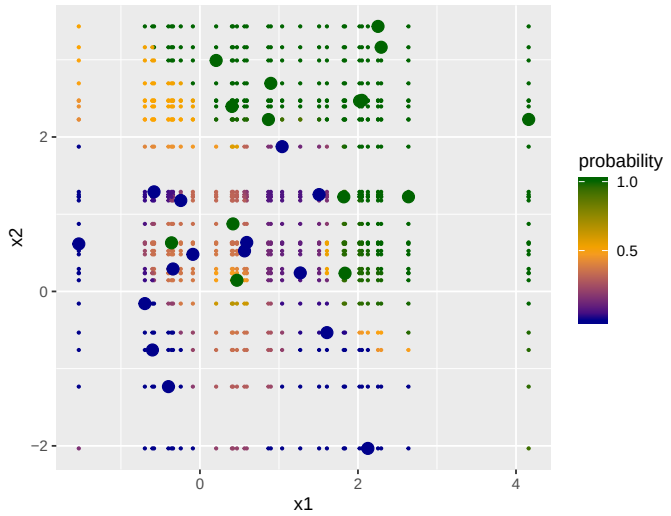

Figure 4: Green large points are from given data with $y = 1$. Blue large points are from $y = -1$. Small dots are calculated conditional probability from the grid.

$$\xi_i \geq 0, \quad i = 1, \ldots, N.$$

If we have the relationship

$$g(\langle H_U X, H_U X \rangle) = \langle H_{U'} h(X), H_{U'} h(X') \rangle \quad \text{and} \quad (\langle X H_V, X H_V \rangle) = \langle h(X) H_{V'}, h(X') H_{V'} \rangle.$$

we can make use of alternating algorithm we do on linear SMM case.

5 Updated rcodes

I updated `svm` function to be available for arbitrary kernel methods.

```
# SVM with kernel functions and weighted cost function
svm = function(X,y,cost = 10, kernels = function(x1,x2) sum(x1*x2), p = .5){
  if (p==.5) {
    cost = 2*cost
  }
  result = list()
  error = 10
  iter = 0
  # SVM
  m= nrow(X[[1]]); n = ncol(X[[1]]); N = length(X)

  x = matrix(unlist(X),nrow = N,byrow = T)
  dvec = rep(1,length(X))
  Dmat = kernelmat(x,y,kernels)
  Amat = cbind(y,diag(1,N),-diag(1,N))
  bvec = c(rep(0,1+N),ifelse(y==1,-cost*(1-p),-cost*p))
  alpha = solve.QP(Dmat,dvec,Amat,bvec,meq =1)
  coef = y*alpha$solution

  Bhat=matrix(t(coef)%*%x,nrow = m)
  # b0hat = -(min(unlist(lapply(X,function(x) sum(Bhat*x)))[which(y==1)])+
  #             max(unlist(lapply(X,function(x) sum(Bhat*x)))[which(y==-1)]))/2
  b0hat = -(min((Dmat%*%alpha$solution)[which(y==1)]*y[which(y==1)])+
              max((Dmat%*%alpha$solution)[which(y==-1)]*y[which(y==-1)]))/2
  obj = objv(Bhat,b0hat,X,y,cost,prob = p)

  predictor = function(y){
    return(sign(sum(coef*apply(x,1, function(x) kernels(x,y)))+b0hat))
  }
  # predictor = function(x) sign(sum(Bhat*x)+b0hat)
  result$B = Bhat; result$b0 = b0hat; result$obj = obj;
  result$predict = predictor
  return(result)
}
```

References

Yi Lin. Support vector machines and the bayes rule in classification. *Data Mining and Knowledge Discovery*, 6:259–275, 2002.

Yi Lin, Yoonkyung Lee, and Grace Wahba. Support vector machines for classification in nonstandard situations. *Machine learning*, 46(1-3):191–202, 2002.