Connection between kernel SMM and SVM

Miaoyan Wang, May 18, 2020

Fact: Let $X \in \mathbb{R}^d$ denote a column vector and $X^* = \begin{bmatrix} 0 & X^T \\ X & 0 \end{bmatrix} \in \mathbb{R}^{(d+1)\times(d+1)}$ be the lifted matrix. Under general nonlinear kernels, we cannot guarantee the equal decision boundaries between X-trained and X^*-trained SMMs!

Where goes wrong? Fact: Repeated attributes are down-weighted in SVM.

Consider two SVMs, one trained on $X = (x_1, x_2)^T$, and another one trained on $X^* = (x_1, x_2, x_2)^T$. It turns out the decision boundary trained by X and X^* are different.

Reason: The primal problems for these two SVMs are

$$(P1) \quad \min_{\beta_1,\beta_2} \left\{ \beta_1^2 + \beta_2^2 + C \sum_i [y_i(\beta_1 x_1 + \beta_2 x_2)]_+ \right\},$$

and

$$\begin{aligned}(P2) \quad & \min_{\beta_1,\beta_2,\beta_3} \left\{ \beta_1^2 + \beta_2^2 + \beta_3^2 + C' \sum_i [y_i(\beta_1 x_1 + (\beta_2 + \beta_3) x_2)]_+ \right\} \\ & = \min_{\beta_1,\beta_2} \left\{ \beta_1^2 + \frac{1}{2}\beta_2^2 + C' \sum_i [y_i(\beta_1 x_1 + \beta_2 x_2)]_+ \right\}.\end{aligned}$$

The solutions to (P1) and (P2) are usually different unless $\beta_1 = 0$. In particular, the repeated attribute, x_2, is down-weighted in the cost function (P2).

Back to the kernel SMM problem. Under general nonlinear kernels, the attributes have different occurrences in $h(X^*)$ and $h(X)$. Therefore, the two decision boundaries might be different. Note that the two classifiers agree in the special case of linear kernels, since each attribute repeats precisely twice.

Implication: A weighted SVM

$$\min_{\boldsymbol{\beta}} \boldsymbol{\beta}^T [\text{Cov}(X)]^{-1} \boldsymbol{\beta} + \sum_i [y_i \langle \boldsymbol{\beta},\ X \rangle]_+$$

will stabilize the contribution from repeated attributes, thereby maintaining the equivalence between X- and X^*-trained SVMs. In our context of kernel SMM, however, the unequal occurrences of attributes are actually okay. We can interpret the unequal occurrences as a prior knowledge of "importance" in the classifier.

Theorem 0.1 (Compatibility with Kernel SVM)**.** *Let $X \in \mathbb{R}^d$ denote a column vector and $X^* =$*

$\begin{bmatrix} 0 & X^T \\ X & 0 \end{bmatrix} \in \mathbb{R}^{(d+1)\times(d+1)}$ be the symmetrized matrix. Then, under our proposed column-wise kernels, the SMM classifier trained on X^* equals the SVM classifier trained on X.

Remark 1. The decision boundaries generated by two classifiers are the same, but the optimal objective values may not be the same.

Proof. We prove the following stronger result.

(**Set-up**.) Let $X \in \mathbb{R}^{m\times n}$ be the original matrix feature,

$$X = \begin{bmatrix} | & | & | & | \\ x_1 & x_2 & \vdots & x_n \\ | & | & | & | \end{bmatrix} = \begin{bmatrix} — & y_1^T & — \\ — & y_2^T & — \\ — & \dots & — \\ — & y_m^T & — \end{bmatrix},$$

where x_i and y_j^T denotes the i-th column and j-th row of the matrix X, respectively. Let $X^* \in \mathbb{R}^{(m+n)\times(m+n)}$ be the symmetrized matrix feature. Based on the assumption (of sufficiently valid kernels), the feature mapping h is applied to matrix X^* in a column-wise fashion; that is,

$$X^* = \begin{bmatrix} 0 & X^T \\ X & 0 \end{bmatrix} \Rightarrow h(X^*) = \left[\begin{array}{c:c} 0 & \begin{matrix} | & | & | \\ h(y_1) & \dots & h(y_m) \\ | & | & | \end{matrix} \\ \hdashline \begin{matrix} | & | & | \\ h(x_1) & \dots & h(x_n) \\ | & | & | \end{matrix} & 0 \end{array}\right] =: \begin{bmatrix} 0 & A \\ C & 0 \end{bmatrix},$$

where, with a little abuse of notation, we have used $h(x_i)$ and $h(y_i)$ to denote the non-zero coordinates in the image vector $h((0,\dots,0,x_i))$ and $h((0,\dots,0,y_j))$, respectively.

(**Conclusion.**) Assume the set of elements in C contains all elements in A. Then, the classifier trained on X equals to the classifier trained on X^*.

(**Proof of the above conclusion.**). Because the elements in C contains all elements in A, we rearrange the elements in C into a possibly larger matrix:

$$C = \begin{bmatrix} 0 & A^T \\ D & 0 \end{bmatrix}.$$

Such arrangement is possible by, for example, setting $D = \text{diag}(\{C\}/\{A\})$. (The element arrangement within D is non-important.) To prove the equivalence between X- and X^*-trained classifiers, it suffices to prove the equivalence between

$$h(X^*) = \begin{bmatrix} 0 & 0 & A \\ 0 & A^T & 0 \\ D & 0 & 0 \end{bmatrix}, \quad \text{and} \quad h(X) = \begin{bmatrix} 0 & A^T \\ D & 0 \end{bmatrix} \quad \text{(c.f. (??) and (??)).}$$

Now we apply the same proof techniques as in 051820_SMMK_modification.pdf. Specifically, consider the primal problems with $h(X^*)$ and $h(X)$. The relevant linear predictors are

$$\langle B^*,\ h(X^*)\rangle = \left\langle \begin{bmatrix} 0 & 0 & B_3^* \\ 0 & B_2^* & 0 \\ B_1^* & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & A \\ 0 & A^T & 0 \\ D & 0 & 0 \end{bmatrix} \right\rangle, \quad \text{and} \quad \langle B,\ X\rangle = \left\langle \begin{bmatrix} 0 & B_2 \\ B_1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & A^T \\ D & 0 \end{bmatrix} \right\rangle.$$

The optimizations are

$$\begin{aligned} (P1) \quad & \min_{B_1^*, B_2^*, B_3^*} \|B_1^*\|_F^2 + \|B_2^*\|_F^2 + \|B_3^*\|_F^2 + C\sum_i \xi_i, \\ & \text{s.t. .... where } \langle B^*,\ h(X^*)\rangle = \langle B_1^*, D\rangle + \langle B_2^*, A^T\rangle + \langle B_3^{*T}, A\rangle \\ & \qquad\qquad\qquad\qquad\qquad = \langle B_1^*, D\rangle + \langle B_2^{*T} + B_3^*, A\rangle, \end{aligned}$$

and

$$\begin{aligned} (P2) \quad & \min_{B_1, B_2} \|B_1\|_F^2 + \|B_2\|_F^2 + C\sum_i \xi_i, \\ & \text{s.t. ... where } \langle B,\ h(X)\rangle = \langle B_1, D\rangle + \langle B_2, A\rangle. \end{aligned}$$

It is easy to see that the optimal solution to (P1) is achieved at $B_2^* = B_3^{*T}$ because

$$\|B_2^*\|_F^2 + \|B_3^*\|_F^2 \geq \frac{1}{2}\|B_2^* + B_3^{*T}\|_F^2.$$

With this choice, the optimizations become

$$(P1) \quad \min_{B_1^*, B_2^*} \|B_1^*\|_F^2 + \frac{1}{2}\|B_2^*\|_F^2 + C\sum_i \left[y_i\left(\langle B_1^*,\ D\rangle + \langle B_2^*,\ A\rangle\right)\right]_+.$$

and

$$(P2) \quad \min_{B_1, B_2} \|B_1\|_F^2 + \|B_2\|_F^2 + C\sum_i \left[y_i\left(\langle B_1,\ D\rangle + \langle B_2,\ A\rangle\right)\right]_+.$$

These two optimizations are different! Repeated features contribute less to (P1) compared to (P2). □