Classification algorithm with matrix kernels

Miaoyan Wang, Aug 17, 2020

Notation:

1. $\mathbb{O}(d,r) := \{\boldsymbol{P} \in \mathbb{R}^{d\times r} \colon \boldsymbol{P}^T\boldsymbol{P} = \boldsymbol{I}_r\}$, the collection of d-by-r matrices whose columns are orthonormal. When no confusion arises, I use the term "projection matrix" to denote either the matrix $\boldsymbol{P}\boldsymbol{P}^T \in \mathbb{R}^{d\times d}$ or the matrix $\boldsymbol{P} \in \mathbb{R}^{d\times r}$.

2. $\mathcal{K}^{\text{row}}(i,j,\boldsymbol{X},\boldsymbol{X'}) := \langle \Phi(\boldsymbol{X}_{i:}),\ \Phi(\boldsymbol{X}'_{j:})\rangle$ denotes the value of row kernel evaluated at the vector pair, (i-th row of matrix $\boldsymbol{X}$, j-th row of matrix $\boldsymbol{X'}$).

3. I sometimes use the shorthand $\mathcal{K}^{\text{row}}(i,j)$ to denote $\mathcal{K}^{\text{row}}(i,j,\boldsymbol{X},\boldsymbol{X'})$, when the feature pair $(\boldsymbol{X},\boldsymbol{X'})$ is clear given the contexts. Note that $\mathcal{K}^{\text{row}}(i,j)$ can be calcualted without explicit feature mapping.

4. Similar convention for $\mathcal{K}^{\text{col}}(i,j,\boldsymbol{X},\boldsymbol{X'})$.

1 Optimization formulation with bilinear mapping

Consider the bilinear mapping,

$$\begin{aligned}\Phi\colon \mathbb{R}^{d_1\times d_2} &\to (\mathcal{H}_r\times\mathcal{H}_c)^{d_1\times d_2}\\ \boldsymbol{X} &\mapsto [\Phi(\boldsymbol{X})_{ij}], \quad \text{where } \Phi(\boldsymbol{X})_{ij} \overset{\text{def}}{=} (\phi_c(\boldsymbol{X}_{i:}),\ \phi_r(\boldsymbol{X}_{:j})).\end{aligned}$$

Primal problem:

$$\begin{aligned}&\min_{\boldsymbol{P}_r,\boldsymbol{P}_c}\min_{\boldsymbol{C}} \quad \frac{1}{2}\|\boldsymbol{C}\|_F^2 + c\sum_{i=1}^n \xi_i, \qquad (1)\\ &\text{subject to} \quad y_i\langle \boldsymbol{P}_r\boldsymbol{C}\boldsymbol{P}_c^T,\ \Phi(\boldsymbol{X}_i)\rangle \le 1-\xi_i \text{ and } \xi_i\ge 0,\ i=1,\ldots,n.\end{aligned}$$

Parameters in the primal problem: $(\boldsymbol{P}_r,\boldsymbol{P}_c,\boldsymbol{C})$, where $\boldsymbol{P}_r \in \mathbb{O}(d_1,r_1)$, $\boldsymbol{P}_c\in\mathbb{O}(d_2,r_2)$, and $\boldsymbol{C} = [\![(\boldsymbol{c}_i^{\text{row}},\ \boldsymbol{c}_j^{\text{col}})]\!] \in (\mathcal{H}_r\times\mathcal{H}_c)^{r_1\times r_2}$ is the low-rank "core matrix" consisting of linear coefficients.

The equivalent dual problem for (1) is

$$\begin{aligned}&\min_{\boldsymbol{P}_r,\boldsymbol{P}_c}\max_{\boldsymbol{\alpha}=(\alpha_1,\ldots,\alpha_n)} \quad \sum_{i=1}^n\alpha_i - \frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_i y_j\langle \boldsymbol{P}_r^T\Phi(\boldsymbol{X}_i)\boldsymbol{P}_c,\ \boldsymbol{P}_r^T\Phi(\boldsymbol{X}_j)\boldsymbol{P}_c\rangle,\\ &\text{subject to} \quad \sum_i y_i\alpha_i = 0,\ \text{and } 0\le\alpha_i\le c,\ i=1,\ldots,n. \qquad (2)\end{aligned}$$

The optimization (2) is also equivalent to

$$\boxed{\begin{aligned}&\max_{\boldsymbol{P}_r,\boldsymbol{P}_c}\min_{\boldsymbol{\alpha}} && -\sum_{i=1}^{n}\alpha_i+\frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_i y_j\langle \boldsymbol{P}_r^T\Phi(\boldsymbol{X}_i)\boldsymbol{P}_c,\ \boldsymbol{P}_r^T\Phi(\boldsymbol{X}_j)\boldsymbol{P}_c\rangle,\\ &\text{subject to} && \sum_i y_i\alpha_i=0,\ \text{and } 0\leq\alpha_i\leq c,\ i=1,\ldots,n,\\ &&& \boldsymbol{P}_r\in\mathbb{O}(d_1,r_1),\ \boldsymbol{P}_c\in\mathbb{O}(d_2,r_2).\end{aligned}} \tag{3}$$

Our goal is to solve (3). The unknown parameters are $(\boldsymbol{P}_r,\boldsymbol{P}_c,\boldsymbol{\alpha})$.

2 Algorithm for problem (3)

1. Update $\boldsymbol{\alpha}$, while holding $(\boldsymbol{P}_r,\boldsymbol{P}_c)$ fixed.

   Prepration: Let $\boldsymbol{W}^{\text{row}}=\boldsymbol{P}_r\boldsymbol{P}_r^T=[\![w_{ij}^{\text{row}}]\!]\in\mathbb{R}^{d_1\times d_1}$ and $\boldsymbol{W}^{\text{col}}=\boldsymbol{P}_c\boldsymbol{P}_c^T=[\![w_{ij}^{\text{col}}]\!]\in\mathbb{R}^{d_2\times d_2}$ denote the row- and column-wise projection matrices, respectively.

   We use kernel trick to solve for $\boldsymbol{\alpha}$ without explicit feature mapping. Given the projections $(\boldsymbol{P}_r,\boldsymbol{P}_c)$, the optimization (3) is a stanard SVM with kernel $\mathcal{K}(\boldsymbol{X},\boldsymbol{X}')$ defined as follows,

$$\begin{aligned}\mathcal{K}(\boldsymbol{X},\boldsymbol{X}')&=\langle \boldsymbol{P}_r^T\Phi(\boldsymbol{X})\boldsymbol{P}_c,\ \boldsymbol{P}_r^T\Phi(\boldsymbol{X}')\boldsymbol{P}_c\rangle\\ &=(\sum_{i,j}w_{ij}^{\text{col}})(\sum_{i,j}w_{ij}^{\text{row}}K^{\text{row}}(i,j))+(\sum_{i,j}w_{ij}^{\text{row}})(\sum_{i,j}w_{ij}^{\text{col}}K^{\text{col}}(i,j)).\end{aligned}\tag{4}$$

   Here I have used the shorthand $K^{\text{row}}(i,j)$ to denote the value of row kernel evaluated on the i-th row of $\boldsymbol{X}$ and j-th row of $\boldsymbol{X}'$.

   Remark 1 (Computational consideration)**.** We can compute the summations in (4) without explicit loop. In particular, both identities hold: $\sum_{i,j}w_{ij}^{\text{col}}=\|\boldsymbol{1}^T\boldsymbol{P}_c\|_2^2$ and $\sum_{i,j}w_{ij}^{\text{row}}K^{\text{row}}(i,j)=\text{trace}(\boldsymbol{W}^T\boldsymbol{K})$, where $\boldsymbol{K}\leftarrow[\![K^{\text{row}}(i,j,\boldsymbol{X},\boldsymbol{X}')]\!]$ is a pre-stored matrix (or array, if we go through all possible feature pairs $(\boldsymbol{X},\boldsymbol{X}')$).

2. Update $\boldsymbol{P}_r$, while holding $(\boldsymbol{\alpha},\boldsymbol{P}_c)$ fixed.

   Denote the matrix $\boldsymbol{M}=\sum_i\alpha_i y_i\Phi(\boldsymbol{X}_i)\boldsymbol{P}_c\in(\mathcal{H}_1\times\mathcal{H}_2)^{d_1\times r_2}$. The problem (3) reduces to

$$\begin{aligned}&\max_{\boldsymbol{P}_r\in\mathbb{O}(d_1,r_1)}\sum_{i,j}\alpha_i\alpha_j y_i y_j\langle \boldsymbol{P}_r^T\Phi(\boldsymbol{X}_i)\boldsymbol{P}_c,\ \boldsymbol{P}_r^T\Phi(\boldsymbol{X}_j)\boldsymbol{P}_c\rangle\\ =&\max_{\boldsymbol{P}_r\in\mathbb{O}(d_1,r_1)}\langle \boldsymbol{P}_r^T\boldsymbol{M},\ \boldsymbol{P}_r^T\boldsymbol{M}\rangle\\ =&\max_{\boldsymbol{P}_r\in\mathbb{O}(d_1,r_1)}\langle\underbrace{\boldsymbol{P}_r\boldsymbol{P}_r^T}_{\text{rank-}r_1\text{ projection}},\ \underbrace{\boldsymbol{M}\boldsymbol{M}^T}_{d_1\text{-by-}d_1\text{ p.s.d. matrix over }\mathbb{R}}\rangle.\end{aligned}\tag{5}$$

By the property of low-rank projection (c.f. Lemma 1), the optimization in the last line has a closed-form solution,

$$\boldsymbol{P}_r \leftarrow \text{top } r_1 \text{ eigenvectors of the matrix } \boldsymbol{M}\boldsymbol{M}^T.$$

It remains to compute the matrix $\boldsymbol{M}\boldsymbol{M}^T$ without explicit feature mapping. Write

$$\begin{aligned}\boldsymbol{M}\boldsymbol{M}^T &= \left(\sum_i \alpha_i y_i \Phi(\boldsymbol{X}_i)\boldsymbol{P}_c\right)\left(\sum_i \alpha_i y_i \Phi(\boldsymbol{X}_i)\boldsymbol{P}_c\right)^T \\ &= \sum_{i,j} \alpha_i\alpha_j y_i y_j \underbrace{\Phi(\boldsymbol{X}_i)\boldsymbol{P}_c\boldsymbol{P}_c^T\Phi^T(\boldsymbol{X}_j)}_{d_1\text{-by-}d_1 \text{ matrix over } \mathbb{R}}. \end{aligned} \tag{6}$$

The summand (6) involves the matrix of the type $\Phi(\boldsymbol{X}_i)\boldsymbol{P}_c\boldsymbol{P}_c^T\Phi^T(\boldsymbol{X}_j)$, for all feature pairs $(i,j) \in [n]^2$. Each of these matrices can be obtained without explicit feature mapping,

$$\begin{aligned}&\Phi(\boldsymbol{X}_i)\boldsymbol{P}_c\boldsymbol{P}_c^T\Phi^T(\boldsymbol{X}_j)\\ &= (\sum_{s,s'} w_{ss'}^{\text{col}}) \begin{bmatrix} K^{\text{row}}(1,1,\boldsymbol{X}_i,\boldsymbol{X}_j) & \cdots & K^{\text{row}}(1,d_1,\boldsymbol{X}_i,\boldsymbol{X}_j) \\ \vdots & \vdots & \vdots \\ K^{\text{row}}(d_1,1,\boldsymbol{X}_i,\boldsymbol{X}_j) & \cdots & K^{\text{row}}(d_1,d_1,\boldsymbol{X}_i,\boldsymbol{X}_j)\end{bmatrix} + \\ &\quad \left(\sum_{s,s'} w_{ss'}^{\text{col}} K^{\text{col}}(s,s',\boldsymbol{X}_i,\boldsymbol{X}_j)\right)\begin{bmatrix} 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & \cdots & 1\end{bmatrix},\end{aligned}$$

where $K^{\text{row}}(s,s',\boldsymbol{X}_i,\boldsymbol{X}_j)$ denotes the value of row kernel value evaluated on the s-th row of $\boldsymbol{X}_i$ and s'-th row of $\boldsymbol{X}_j$, and likewise for $K^{\text{col}}(s,s',\boldsymbol{X}_i,\boldsymbol{X}_j)$.

3. Update $\boldsymbol{P}_c$, while holding $(\boldsymbol{\alpha}, \boldsymbol{P}_r)$ fixed. Similar as step 2 but switching the role of rows and columns.

Lemma 1 (Best rank-r projection)**.** *Let $\boldsymbol{A} \in \mathbb{R}^{d\times d}$ be a positive semi-definite matrix. Let $(\lambda_i, \boldsymbol{p}_i) \in \mathbb{R}\times\mathbb{R}^d$ denote the i-th singular-value-singularvector pair of $\boldsymbol{A}$, and assume that eigenvalues $\lambda_1 \geq \cdots \geq \lambda_d \geq 0$ are sorted in non-increasing order. Consider an optimization problem specified as*

$$\max_{\boldsymbol{P}\in\mathbb{O}(d,r)} f(\boldsymbol{P}), \quad \textit{where} \quad f(\boldsymbol{P}) = \langle \boldsymbol{P}\boldsymbol{P}^T,\ \boldsymbol{A}\rangle.$$

Then, the leading rank-r singular space of $\boldsymbol{A}$, denoted $\boldsymbol{P}^* = Span(\boldsymbol{p}_1,\ldots,\boldsymbol{p}_r)$, optimizes the objective $f(\boldsymbol{P})$. In particular, $f(\boldsymbol{P}^*) = \sum_{i=1}^r \lambda_i(\boldsymbol{A})$.

Proof. The positive semi-definiteness of $\boldsymbol{A}$ implies the existence of a symmetric matrix $\boldsymbol{B} \in \mathbb{R}^{d\times d}$ such that $\boldsymbol{A} = \boldsymbol{B}^2$. Furthermore, the singular values satisfy $\lambda_i^2(\boldsymbol{B}) = \lambda_i(\boldsymbol{A})$ for all $i \in [d]$. Notice that

$$f(\boldsymbol{P}) = \langle \boldsymbol{P}\boldsymbol{P}^T,\ \boldsymbol{B}^2\rangle = \|\boldsymbol{B}\|_F^2 - \|\underbrace{\boldsymbol{B}(\boldsymbol{I}-\boldsymbol{P}\boldsymbol{P}^T)}_{\text{rank-}(d-r)\text{ approximation of }\boldsymbol{B}}\|_F^2 \leq \sum_{i=1}^r \lambda_i^2(\boldsymbol{B})$$

holds for all matrices $\boldsymbol{P} \in \mathbb{O}(d,r)$. Therefore,

$$\max_{\boldsymbol{P}\in\mathbb{O}(d,r)} f(\boldsymbol{P}) \leq \sum_{i=1}^r \lambda_i(\boldsymbol{A}),$$

where equality is attained if $\boldsymbol{P} = \text{Span}(\boldsymbol{p}_1,\ldots,\boldsymbol{p}_r)$. □

3 Outputs

How to read off the decision function from the algorithm outputs?

$$\begin{aligned} f(\boldsymbol{X}_{\text{new}}) &= \langle \boldsymbol{P}_r^T\Phi(\boldsymbol{X}_{\text{new}})\boldsymbol{P}_c,\ \sum_i \alpha_i y_i \boldsymbol{P}_r^T\Phi(\boldsymbol{X}_i)\boldsymbol{P}_c\rangle \\ &= \langle \Phi(\boldsymbol{X}_{\text{new}}),\ \boldsymbol{P}_r \underbrace{\boldsymbol{P}_r^T\left(\sum_i \alpha_i y_i \Phi(\boldsymbol{X}_i)\right)\boldsymbol{P}_c}_{\text{core tensor }\boldsymbol{C}\text{ in the primal problem}} \boldsymbol{P}_c^T\rangle \\ &= \sum_i \alpha_i y_i \Bigg\{ \left(\sum_{s,s'} w_{ss'}^{\text{col}}\right)\left(\sum_{s,s'} w_{ss'}^{\text{row}} K^{\text{row}}(s,s',\boldsymbol{X}_i,\boldsymbol{X}_{\text{new}})\right) + \\ &\qquad \left(\sum_{s,s'} w_{ss'}^{\text{row}}\right)\left(\sum_{s,s'} w_{ss'}^{\text{col}} K^{\text{col}}(s,s',\boldsymbol{X}_i,\boldsymbol{X}_{\text{new}})\right)\Bigg\}. \end{aligned} \tag{7}$$

How to estimate the intercept in the primal problem?

$$\hat{b}_0 = \operatorname*{arg\,min}_{b_0\in\mathbb{R}} \left\{ \frac{1}{2}\|\boldsymbol{C}\|_F^2 + c\sum_{i=1}^n (1 - y_i f(\boldsymbol{X}_i) - y_i b_0)_+ \right\},$$

where $\|\boldsymbol{C}\|_F^2 = \sum_{i,j} \alpha_i\alpha_j y_i y_j \langle \boldsymbol{P}_r^T\Phi(\boldsymbol{X}_i)\boldsymbol{P}_c,\ \boldsymbol{P}_r^T\Phi(\boldsymbol{X}_j)\boldsymbol{P}_c\rangle = \sum_{i=1}^r \lambda_i(\boldsymbol{M}\boldsymbol{M}^T)$, and $\lambda_i(\cdot)$ denotes the i-th eigenvalue of the matrix. The formula for $\|\boldsymbol{C}\|_F^2$ follows from the second line of (7) and the optimization (5).

4 Further thoughts

The dual optimization (3) yields a neater algorithm than previous approaches. Recall that, in the notes *0423.pdf and *0620.pdf, we have derived the alternating optimization algorithm for the primal problem,

$$
\boxed{
\begin{aligned}
&\min_{\boldsymbol{P}} \min_{\boldsymbol{C}} \quad \frac{1}{2}\|\boldsymbol{C}\boldsymbol{P}^T\|_F^2 + c\sum_{i=1}^n \xi_i,\\
&\text{subject to} \quad y_i\langle \boldsymbol{C}\boldsymbol{P}^T, \Phi(\boldsymbol{X}_i)\rangle \leq 1-\xi_i \text{ and } \xi_i \geq 0, i=1,\ldots,n,\\
&\qquad\qquad \text{where } \boldsymbol{C}=(\boldsymbol{C}_r,\boldsymbol{C}_c),\ \boldsymbol{P}=(\boldsymbol{P}_r,\boldsymbol{P}_c)\in\mathbb{O}(d_1,r)\times\mathbb{O}(d_2,r).
\end{aligned}
}
\tag{8}
$$

The block variable $\boldsymbol{P}$ has explicit update, whereas the other block $\boldsymbol{C}$ has implicit update. Here we give a different perspective on the algorithm derivation. Notice that the primal problem (8) is equivalent to the dual problem,

$$
\boxed{
\begin{aligned}
&\max_{\boldsymbol{P}} \min_{\boldsymbol{\alpha}} \quad -\sum_{i=1}^n \alpha_i + \frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_i y_j \langle \Phi(\boldsymbol{X}_i)\boldsymbol{P},\ \Phi(\boldsymbol{X}_j)\boldsymbol{P}\rangle,\\
&\text{subject to} \quad \sum_i y_i\alpha_i = 0,\ \text{and } 0\leq \alpha_i \leq c,\ i=1,\ldots,n,\\
&\qquad\qquad \boldsymbol{P}=(\boldsymbol{P}_r,\ \boldsymbol{P}_c)\in\mathbb{O}(d_1,r_1)\times\mathbb{O}(d_2,r_2).
\end{aligned}
}
\tag{9}
$$

(For notational convenience, I will drop the column-wise projection $\boldsymbol{P}_c$ and consider row-wise projection $\boldsymbol{P}_r$ only. In such a case, $\Phi(\boldsymbol{X})\in\mathcal{H}^{d_1}$.)

Algorithm for optimization (9) over parameters $(\boldsymbol{P},\boldsymbol{\alpha})$.

1. Update $\boldsymbol{\alpha}$ holding $\boldsymbol{P}$ fixed. $\Longrightarrow$ same as in the note *0620.pdf.

2. Update $\boldsymbol{P}$ holding $\boldsymbol{\alpha}$ fixed.

$$
\begin{aligned}
\boldsymbol{P} &\leftarrow \mathop{\arg\max}_{\boldsymbol{P}\in\mathbb{O}(d,r)} \sum_{i,j}\alpha_i\alpha_j y_i y_j\langle\Phi(\boldsymbol{X}_i)\boldsymbol{P},\ \Phi(\boldsymbol{X}_j)\boldsymbol{P}\rangle\\
&\overset{\text{c.f. Lemma 1}}{=} \text{top } r \text{ singular vectors of matrix } \boldsymbol{B}\boldsymbol{B}^T, \quad \text{where } \boldsymbol{B}=\underbrace{\sum_{i=1}^n \alpha_i y_i \Phi(\boldsymbol{X}_i)}_{\mathcal{H}^{d_1}}.
\end{aligned}
$$

Notice that $\boldsymbol{B}\boldsymbol{B}^T$ can be obtained without explicit feature mapping,

$$
\boldsymbol{B}\boldsymbol{B}^T = \left(\sum_{i=1}^n \alpha_i y_i \Phi(\boldsymbol{X}_i)\right)\left(\sum_{i=1}^n \alpha_i y_i \Phi(\boldsymbol{X}_i)\right)^T = \sum_{i,j}\alpha_i\alpha_j y_i y_j \Phi(\boldsymbol{X}_i)\Phi^T(\boldsymbol{X}_j).
$$

As a by-product, the dual formulation (9) also justifies the same treatment to coefficients $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ in the previous algorithm.

Remark 2. In theory, alternating optimization may not solve the general minmax problem (9). In practice perhaps okay? Does the objective converge over iterations? Need to check.