Algorithm for bilinear maps

Miaoyan Wang, Aug 21, 2020

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

2 Algorithm based on bilinear maps

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:

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

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 "core matrix" consisting of linear coefficients.

1. Update $\boldsymbol{C}$, given $(\boldsymbol{P}_r,\boldsymbol{P}_c)$.

$$\boxed{\text{implicit update } \boldsymbol{C} \leftarrow \sum_i \alpha_i y_i \boldsymbol{P}_r^T\Phi(\boldsymbol{X}_i)\boldsymbol{P}_c.}$$

- Saved quantities: dual variables $\boldsymbol{\alpha} \in \mathbb{R}^n$.
- Auxiliary quantities: kernel $\mathcal{K}(\boldsymbol{X}, \boldsymbol{X}')$ specified below.
- Objective value: objective value for the dual problem.

Details: We use kernel trick to solve for $\boldsymbol{\alpha}$ without explicit feature mapping. Given the projections $(\boldsymbol{P}_r, \boldsymbol{P}_c)$, the optimization (5) is a standard SVM with kernel

$$\begin{aligned}\mathcal{K}(\boldsymbol{X}, \boldsymbol{X}') &= \langle \boldsymbol{P}_r^T \Phi(\boldsymbol{X}) \boldsymbol{P}_c,\ \boldsymbol{P}_r^T \Phi(\boldsymbol{X}') \boldsymbol{P}_c \rangle \\ &= (\sum_{i,j} w_{ij}^{\text{col}})(\sum_{i,j} w_{ij}^{\text{row}} K^{\text{row}}(i,j)) + (\sum_{i,j} w_{ij}^{\text{row}})(\sum_{i,j} w_{ij}^{\text{col}} K^{\text{col}}(i,j)).\end{aligned} \tag{2}$$

for all feature pairs $(\boldsymbol{X}, \boldsymbol{X}')$. 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 (2) without explicit loop. In particular, both identities hold: $\sum_{i,j} w_{ij}^{\text{col}} = \|\mathbf{1}^T \boldsymbol{P}_c\|_2^2$ 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 $\boldsymbol{P}_r^{\text{new}}$, given $(\boldsymbol{C}, \boldsymbol{P}_c)$.

$$\boxed{\begin{aligned}&\text{explicit update } \tilde{\boldsymbol{P}}_r^{\text{new}} \leftarrow \sum_i \beta_i y_i \Phi(\boldsymbol{X}_i) \boldsymbol{P}_c \boldsymbol{C}^T = \sum_{i,j} \beta_i \alpha_j y_i y_j \underbrace{\Phi(\boldsymbol{X}_i) \boldsymbol{P}_c \boldsymbol{P}_c^T \Phi^T(\boldsymbol{X}_j)}_{d_1\text{-by-}d_1 \text{ matrix over } \mathbb{R}} \boldsymbol{P}_r \\ &\qquad \text{normalize } \boldsymbol{P}_r^{\text{new}} \leftarrow \text{QR decomposition of } \tilde{\boldsymbol{P}}_r^{\text{new}}.\end{aligned}}$$

- Saved quantities: $\boldsymbol{P}^{\text{new}} \in \mathbb{O}(d_1, r_1)$.
- Auxiliary quantities: matrix $\Phi(\boldsymbol{X}_i) \boldsymbol{P}_c \boldsymbol{P}_c^T \Phi^T(\boldsymbol{X}_j)$, dual variables $\boldsymbol{\beta} \in \mathbb{R}^n$.
- Objective value: objective value for the dual problem.

Details: for each feature pair $(i,j) \in [n]^2$, we compute the matrix $\Phi(\boldsymbol{X}_i) \boldsymbol{P}_c \boldsymbol{P}_c^T \Phi^T(\boldsymbol{X}_j)$ without explicit feature mapping,

$$\begin{aligned}\underbrace{\Phi(\boldsymbol{X}_i) \boldsymbol{P}_c \boldsymbol{P}_c^T \Phi^T(\boldsymbol{X}_j)}_{=:\boldsymbol{M}} = (\sum_{s,s'} w_{ss'}^{\text{col}}) &\begin{bmatrix} K^{\text{row}}(1,1,\boldsymbol{X}_i,\boldsymbol{X}_j) & \cdots & K^{\text{row}}(1,d_1,\boldsymbol{X}_i,\boldsymbol{X}_j) \\ \vdots & \vdots & \vdots \\ K^{\text{row}}(d_1,1,\boldsymbol{X}_i,\boldsymbol{X}_j) & \cdots & K^{\text{row}}(d_1,d_1,\boldsymbol{X}_i,\boldsymbol{X}_j) \end{bmatrix} + \\ &\left(\sum_{s,s'} w_{ss'}^{\text{col}} K^{\text{col}}(s,s',\boldsymbol{X}_i,\boldsymbol{X}_j) \right) \begin{bmatrix} 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & \cdots & 1 \end{bmatrix},\end{aligned} \tag{3}$$

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 $K^{\text{col}}(s,s',\boldsymbol{X}_i,\boldsymbol{X}_j)$.

The coefficient $\boldsymbol{\beta}$ is obtained from a standard SVM with kernel

$$\mathcal{K}(\boldsymbol{X}, \boldsymbol{X}') = \text{trace}(\underbrace{\Phi(\boldsymbol{X})\boldsymbol{P}_c\boldsymbol{C}^T}_{=:\boldsymbol{A}}(\underbrace{\boldsymbol{C}\boldsymbol{C}^T}_{=:\boldsymbol{B}})^{-1}\underbrace{\boldsymbol{C}\boldsymbol{P}_c^T\Phi^T(\boldsymbol{X}')}_{=:\boldsymbol{A}^T}),$$

for feature pair $(\boldsymbol{X}, \boldsymbol{X}')$, where

$$\begin{aligned}\boldsymbol{B} = \boldsymbol{C}\boldsymbol{C}^T &= \sum_{i,j}\alpha_i\alpha_j y_i y_j \boldsymbol{P}_r^T \Phi(\boldsymbol{X}_i)\boldsymbol{P}_c\boldsymbol{P}_c^T\Phi^T(\boldsymbol{X}_j)\boldsymbol{P}_r,\\ \boldsymbol{A} = \Phi(\boldsymbol{X})\boldsymbol{P}_c\boldsymbol{C}^T &= \sum_i \alpha_i y_i \Phi(\boldsymbol{X})\boldsymbol{P}_c\boldsymbol{P}_c^T\Phi^T(\boldsymbol{X}_i)\boldsymbol{P}_r.\end{aligned}$$

are calculated using (3).

3. Update $\boldsymbol{C}^{\text{new}}$, given $(\boldsymbol{P}_r^{\text{new}}, \boldsymbol{P}_c)$.

4. Update $\boldsymbol{P}_c$, given $(\boldsymbol{C}^{\text{new}}, \boldsymbol{P}_r^{\text{new}})$; denoted by $(\boldsymbol{C}, \boldsymbol{P}_r)$ in the following formula.

$$\boxed{\begin{aligned}&\text{explicite update } \boldsymbol{P}_c^{\text{new}} \leftarrow \sum_i \gamma_i y_i \Phi^T(\boldsymbol{X}_i)\boldsymbol{P}_r\boldsymbol{C} = \sum_{i,j}\gamma_i\alpha_j y_i y_j \underbrace{\Phi^T(\boldsymbol{X}_i)\boldsymbol{P}_r\boldsymbol{P}_r^T\Phi(\boldsymbol{X}_j)}_{d_2\text{-by-}d_2 \text{ matrix over } \mathbb{R}}\boldsymbol{P}_c\\ &\qquad\text{normalize } \boldsymbol{P}_c^{\text{new}} \leftarrow \text{QR decomposition of } \tilde{\boldsymbol{P}}_c^{\text{new}}.\end{aligned}}$$

The auxiliary quantities, $\Phi^T(\boldsymbol{X}_i)\boldsymbol{P}_r\boldsymbol{P}_r^T\Phi(\boldsymbol{X}_j)$ and $\boldsymbol{\gamma}$, are calculated similarly as in step 2.

3 Outputs

1. Convergence criterum? Objective value in the dual problem.

2. How to read off the decision function from the algorithm?

$$\begin{aligned}f(\boldsymbol{X}_{\text{new}}) &= \langle \boldsymbol{P}_r^T\Phi(\boldsymbol{X}_{\text{new}})\boldsymbol{P}_c, \ \sum_i \alpha_i y_i \boldsymbol{P}_r^T\Phi(\boldsymbol{X}_i)\boldsymbol{P}_c\rangle\\ &= \sum_i \alpha_i y_i \left\{ \left(\sum_{s,s'} w_{ss'}^{\text{col}}\right)\left(\sum_{s,s'} w_{ss'}^{\text{row}} K^{\text{row}}(s,s',\boldsymbol{X}_i,\boldsymbol{X}_{\text{new}})\right) + \right.\\ &\qquad\left. \left(\sum_{s,s'} w_{ss'}^{\text{row}}\right)\left(\sum_{s,s'} w_{ss'}^{\text{col}} K^{\text{col}}(s,s',\boldsymbol{X}_i,\boldsymbol{X}_{\text{new}})\right)\right\}.\end{aligned}$$

3. How to read off the low-rank coefficients from the linear kernel?

$$f(\boldsymbol{X}_{\text{new}}) = (\sum_{s,s'} w_{ss'}^{\text{col}})\langle \boldsymbol{B}^{\text{row}}, \boldsymbol{X}_{\text{new}}\rangle + (\sum_{s,s'} w_{ss'}^{\text{row}})\langle \boldsymbol{B}^{\text{col}}, \boldsymbol{X}_{\text{new}}\rangle,$$

where

$$\boldsymbol{B}^{\text{row}} = \boldsymbol{P}^{\text{row}}(\boldsymbol{P}^{\text{row}})^T(\sum_i \alpha_i y_i \boldsymbol{X}_i) \quad \text{and} \quad \boldsymbol{B}^{\text{col}} = (\sum_i \alpha_i y_i \boldsymbol{X}_i)\boldsymbol{P}^{\text{col}}(\boldsymbol{P}^{\text{col}})^T.$$

Upon convergence we should have $\boldsymbol{B}^{\text{row}} = \boldsymbol{B}^{\text{col}}$???

Lemma 1. *Consider an optimization problem with penalty parameter $c > 0$,*

$$\min_{rank(\boldsymbol{B}_1)+rank(\boldsymbol{B}_2)\leq r} f(\boldsymbol{B}_1 + \boldsymbol{B}_2) + c(\|\boldsymbol{B}_1\|_F^2 + \|\boldsymbol{B}_2\|_F^2). \tag{4}$$

Suppose $(\boldsymbol{B}_1^*, \boldsymbol{B}_2^*)$ is optimal solution to the optimization (8). Then, we must have $\boldsymbol{B}_1^* = \boldsymbol{B}_2^*$.

Proof. Define a new solution $(\tilde{\boldsymbol{B}}_1^*, \tilde{\boldsymbol{B}}_2^*) = (\boldsymbol{B}^*/2, \boldsymbol{B}^*/2)$, where $\boldsymbol{B}^* = \boldsymbol{B}_1^* + \boldsymbol{B}_2^*$. Note that the new solution is in the feasible decision set, $\text{rank}(\tilde{\boldsymbol{B}}_1^* + \tilde{\boldsymbol{B}}_2^*) = \text{rank}(\boldsymbol{B}^*) \leq r$. Furthermore, the optimality of $(\boldsymbol{B}_1^*, \boldsymbol{B}_2^*)$ implies

$$f(\boldsymbol{B}_1^*, \boldsymbol{B}_2^*) + c(\|\boldsymbol{B}_1^*\|_F^2 + \|\boldsymbol{B}_1^*\|_F^2) \leq f(\tilde{\boldsymbol{B}}_1^* + \tilde{\boldsymbol{B}}_2^*) + c(\|\tilde{\boldsymbol{B}}_1^*\|_F^2 + \|\tilde{\boldsymbol{B}}_2^*\|_F^2). \tag{5}$$

Plugging the definition $\tilde{\boldsymbol{B}}_1^* = \tilde{\boldsymbol{B}}_2^* = \frac{1}{2}(\boldsymbol{B}_1^* + \boldsymbol{B}_2^*)$ into (5), we have

$$\|\boldsymbol{B}_1^*\|_F^2 + \|\boldsymbol{B}_1^*\|_F^2 \leq \frac{1}{2}\|\boldsymbol{B}_1^* + \boldsymbol{B}_2^*\|_F^2. \tag{6}$$

The proof is completed by noting that the inequality (6) holds only if $\boldsymbol{B}_1^* = \boldsymbol{B}_2^*$. □

Remark 2. Lemma (1) implies that the solution to symmetrized algorithm must be of

$$\boldsymbol{B} = \begin{bmatrix} \boldsymbol{C} & \boldsymbol{B}_1 \\ \boldsymbol{B}_2^T & \boldsymbol{D} \end{bmatrix}.$$

Under the linear kernel, $f(\boldsymbol{B}) = f(\boldsymbol{B}_1 + \boldsymbol{B}_2)$. Furthermore, $\|\boldsymbol{B}\|_F \geq \|\boldsymbol{B}_1\|_F^2 + \|\boldsymbol{B}_2\|_F^2$.

Lemma 2. *Let $(\boldsymbol{B}_1^*, \boldsymbol{B}_2^*)$ be the optimal solution to the following problem*

$$\min_{rank(\boldsymbol{B}_1)\leq r,\ rank(\boldsymbol{B}_2)\leq r} f(\boldsymbol{B}_1 + \boldsymbol{B}_2) + c(\|\boldsymbol{B}_1\|_F^2 + \|\boldsymbol{B}_2\|_F^2). \tag{7}$$

Then, we must have $\boldsymbol{B}_1^* = \boldsymbol{B}_2^*$.

Proof. Define a new solution $(\tilde{\boldsymbol{B}}_1^*, \tilde{\boldsymbol{B}}_2^*) = (\frac{\boldsymbol{B}_1^*+\boldsymbol{B}_2^*}{2}, \frac{\boldsymbol{B}_1^*+\boldsymbol{B}_2^*}{2})$. Note that $\text{rank}(\boldsymbol{B}_1^*) = \text{rank}(\boldsymbol{B}_1^*) \leq 2r$. Therefore

$$f(\boldsymbol{B}_1^* + \boldsymbol{B}_2^*) + c(\|\boldsymbol{B}_1^*\|_F^2 + \|\boldsymbol{B}_2^*\|_F^2) \leq f(\boldsymbol{B}_1 + \boldsymbol{B}_2) + c(\|\boldsymbol{B}_1\|_F^2 + \|\boldsymbol{B}_2\|_F^2).$$

$$f(\boldsymbol{B}_1 + \boldsymbol{B}_2) + \frac{c}{2}\|\boldsymbol{B}_1 + \boldsymbol{B}_2\|_F^2 \leq f(\boldsymbol{B}_1 + \boldsymbol{B}_2) + c(\|\boldsymbol{B}_1\|_F^2 + \|\boldsymbol{B}_2\|_F^2).$$

$$\|\boldsymbol{B}_1 + \boldsymbol{B}_2\|_F^2 \leq 2(\|\boldsymbol{B}_1\|_F^2 + \|\boldsymbol{B}_2\|_F^2).$$

□

$$\min_{rank(\boldsymbol{B}_1+\boldsymbol{B}_2)\leq r} f(\boldsymbol{B}_1 + \boldsymbol{B}_2) + c(\|\boldsymbol{B}_1\|_F^2 + \|\boldsymbol{B}_2\|_F^2). \tag{8}$$

The optimal solution to (8) must satisfy $\boldsymbol{B}_1^* = \boldsymbol{B}_2^*$.

4. How to estimate the intercept in the primal problem?

$$\hat{b}_0 = \arg\min_{b_0\in\mathbb{R}} \left\{ \sum_{i=1}^{n} (1 - y_i f(\boldsymbol{X}_i) - y_i b_0)_+ \right\}.$$

5. Penalty in the primal problem?

$$\|\boldsymbol{C}\|_F^2 = \sum_{i,j} \alpha_i\alpha_j y_i y_j \text{trace}(\Phi(\boldsymbol{X}_i)\boldsymbol{P}_c\boldsymbol{P}_c^T\Phi^T(\boldsymbol{X}_j)\boldsymbol{P}_r\boldsymbol{P}_r^T) = (\boldsymbol{\alpha}\circ\boldsymbol{y})^T [\![\mathcal{K}(\boldsymbol{X}_i,\boldsymbol{X}_j)]\!](\boldsymbol{\alpha}\circ\boldsymbol{y}),$$

where $[\![\mathcal{K}(\boldsymbol{X}_i,\boldsymbol{X}_j)]\!] \in \mathbb{R}^{n\times n}$ is defined in (2) and $\boldsymbol{\alpha} \in \mathbb{R}^n$ is the dual parameter in step 1 or 3. We omit the explicit expression of $\|\boldsymbol{C}\|_F^2$ because it is not needed in our algorithm.

4 Sanity check

1. Monotonic decreasing in objective value? Yes.
2. Consistent with earlier linear kernel algorithm, smmk(....)? Yes.
3. Consistent with one-way linear mapping? Yes.

```
## generate data with linear kernel
source("SMMfunctions.R")
set.seed(1818)
m = 3; n = 3; r = 1; N = 100; b0 = 0.1
result = gendat(m,n,r,N,b0)
X = result$X; y = result$y; dat = result$dat;B = result$B
X_t=X ## transform
y_true=rep(0,N)
for(i in 1:N){
X_t[[i]]=t(X[[i]])
y_true[i]=sum(B*X[[i]])
}
svd(B)$v[,1] ## true col projection
svd(B)$u[,1] ## true row projection
```

```
## Algorithm 1: earlier algorithm
source("SMMKfunctions.R")
old= smmk(X,y,r=1,kernel = "linear");
old$V ## est col projection
old_t= smmk(X_t,y,r=1,kernel = "linear");
old_t$V ## est row projection

## Algorithm 2: one-way map. Separate row/column projection.
source("SMMKfunctions_bilinear.R")
new = smmk_new(X,y,kernel_row="const",kernel_col="linear",
r=c(1,1),cost=10/3);
new$P_col ## est col projection; similar to earlier old$V
new = smmk_new(X,y,kernel_row="linear",kernel_col="const",
r=c(1,1),cost=10/3);
new$P_row ## est row projection; similar to earlier old_t$V


### Algorithm 2: two-way map. Simultaneous row/column projections.
new_double = smmk_new(X,y,kernel_row="linear",kernel_col="linear",
r=c(1,1),cost=10,rep=1);
plot(new_double$obj)
new_double$P_col  ## est col projection
new_double$P_row ## est row projection
plot(new_double$fitted,y_true)

### Algorithm 2: two-way map + transform. Results should be consistent.
new_double  = smmk_new(X_t,y,kernel_row="linear",kernel_col="linear",
r=c(1,1),cost=10,rep=1);
new_double$P_row ## est col projection
new_double$P_col ## est row projeciotn
plot(new_double$fitted,y_true)
```

Sanity check under linear kernel, $d = 3$, $r = 1$, $N = 100$.

Parameters	Truth	Symmetric trick (SMMK)	Bilinear map
col projection	$(0.63, 0.33, 0.70)$	$(0.66, 0.33, 0.67)$	$(0.70, 0.39, 0.58)$
			$(0.64, 0.53, 0.55)$
row projection	$(-0.54, 0.83, 0.08)$	$(-0.55, 0.83, 0.06)$	$(-0.48, 0.87, 0.09)$
			$(-0.53, 0.84, 0.07)$