Supplementary material of online high-rank matrix completion

Jicong Fan, Madeleine Udell
Cornell University
Ithaca, NY 14853, USA
{jf577,udell}@cornell.edu

1. Proof of some lemmas

1.1. Proof for Lemma 3

Lemma 3. *The update (8) is a relaxed Newton's method and ensures sufficient decrease in the objective:*

$$\ell(\boldsymbol{Z},\boldsymbol{D}-\boldsymbol{\Delta}_D,\boldsymbol{X})-\ell(\boldsymbol{Z},\boldsymbol{D},\boldsymbol{X})\leq -\tfrac{1}{2\tau}\mathrm{Tr}(\boldsymbol{g}_D\boldsymbol{H}_D^{-1}\boldsymbol{g}_D^\top).$$

Proof. With polynomial kernel, the objective function in terms of $\boldsymbol{D}$ is

$$\begin{aligned}\ell(\boldsymbol{Z},\boldsymbol{D},\boldsymbol{X}) =& -\mathrm{Tr}((\boldsymbol{W}_1\odot(\boldsymbol{X}^T\boldsymbol{D}+c))\boldsymbol{Z})\\ &+\tfrac{1}{2}\mathrm{Tr}(\boldsymbol{Z}^T(\boldsymbol{W}_2\odot(\boldsymbol{D}^T\boldsymbol{D}+c))\boldsymbol{Z})\\ &+\tfrac{\alpha}{2}\mathrm{Tr}(\boldsymbol{W}_2\odot(\boldsymbol{D}^T\boldsymbol{D}+c)),\end{aligned}\tag{1}$$

in which for simplicity we have omitted the terms not related to $\boldsymbol{D}$. In (1), $\boldsymbol{W}_1=\langle\boldsymbol{X}^T\boldsymbol{D}+c\rangle^{q-1}$, $\boldsymbol{W}_2=\langle\boldsymbol{D}^T\boldsymbol{D}+c\rangle^{q-1}$, and $\langle\cdot\rangle^u$ denotes the element-wise u-power of a vector or matrix. Using the idea of iteratively reweighted optimization, we fix $\boldsymbol{W}_1$ and $\boldsymbol{W}_2$, and get the derivative as

$$\boldsymbol{g}_D:=-\boldsymbol{X}(\boldsymbol{W}_1\odot\boldsymbol{Z}^T)+\boldsymbol{D}(\boldsymbol{Z}\boldsymbol{Z}^T\odot\boldsymbol{W}_2)+\alpha\boldsymbol{D}(\boldsymbol{W}_2\odot\boldsymbol{I}_r)).\tag{2}$$

We approximate $\ell(\boldsymbol{Z},\boldsymbol{D},\boldsymbol{X})$ with its second order Taylor expansion around $\boldsymbol{D}_0$, i.e.,

$$\begin{aligned}\ell(\boldsymbol{Z},\boldsymbol{D},\boldsymbol{X}) =&\ell(\boldsymbol{Z},\boldsymbol{D}_0,\boldsymbol{X})+\langle\boldsymbol{g}_D,\boldsymbol{D}-\boldsymbol{D}_0\rangle\\ &+\tfrac{1}{2}\mathrm{vec}(\boldsymbol{D}-\boldsymbol{D}_0)^T\boldsymbol{H}_D\mathrm{vec}(\boldsymbol{D}-\boldsymbol{D}_0)+R_0,\end{aligned}\tag{3}$$

where $R_0=O(\frac{\|\ell^{(3)}\|}{6})$ denotes the residual of the approximation and $\boldsymbol{H}\in\mathbb{R}^{r^2\times r^2}$ denotes the Hessian matrix. We have

$$\boldsymbol{H}=\begin{bmatrix}\boldsymbol{H}_D & \boldsymbol{0} & \cdots & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{H}_D & \cdots & \boldsymbol{0}\\ \vdots & \cdots & \ddots & \vdots\\ \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{H}_D\end{bmatrix},\tag{4}$$

where

$$\boldsymbol{H}_D:=\boldsymbol{Z}\boldsymbol{Z}^T\odot\boldsymbol{W}_2+\alpha\boldsymbol{W}_2\odot\boldsymbol{I}_r.\tag{5}$$

One has $\mathrm{vec}(\boldsymbol{D}-\boldsymbol{D}_0)^T\boldsymbol{H}\mathrm{vec}(\boldsymbol{D}-\boldsymbol{D}_0)=\mathrm{Tr}((\boldsymbol{D}-\boldsymbol{D}_0)\boldsymbol{H}(\boldsymbol{D}-\boldsymbol{D}_0)^T)$. Denote

$$\begin{aligned}\ell'(\boldsymbol{Z},\boldsymbol{D},\boldsymbol{X}) =&\ell(\boldsymbol{Z},\boldsymbol{D}_0,\boldsymbol{X})+\langle\boldsymbol{g}_D,\boldsymbol{D}-\boldsymbol{D}_0\rangle\\ &+\tfrac{\tau}{2}\mathrm{Tr}((\boldsymbol{D}-\boldsymbol{D}_0)\boldsymbol{H}_D(\boldsymbol{D}-\boldsymbol{D}_0)^T),\end{aligned}\tag{6}$$

where $\tau>1$. Since $\boldsymbol{H}_D$ is positive definite, we have

$$\ell(\boldsymbol{Z},\boldsymbol{D},\boldsymbol{X})\leq\ell'(\boldsymbol{Z},\boldsymbol{D},\boldsymbol{X}),\tag{7}$$

provided that τ is large enough. We then minimize ℓ' by letting the derivative be zero and get

$$\boldsymbol{D}=\boldsymbol{D}_0-\boldsymbol{\Delta}_D.\tag{8}$$

where $\boldsymbol{\Delta}_D=\frac{1}{\tau}\boldsymbol{g}_D\boldsymbol{H}_D^{-1}$. Invoking (8) into (7), we have

$$\ell(\boldsymbol{Z},\boldsymbol{D}_0-\boldsymbol{\Delta}_D,\boldsymbol{X})\leq\ell(\boldsymbol{Z},\boldsymbol{D}_0,\boldsymbol{X})-\tfrac{1}{2\tau}\mathrm{Tr}(\boldsymbol{g}_D\boldsymbol{H}_D^{-1}\boldsymbol{g}_D^T).\tag{9}$$

□

1.2. Proof for Lemma 4

Lemma 4. $\|\boldsymbol{X}(\boldsymbol{Z}^T\odot\boldsymbol{K}_{\boldsymbol{X}\boldsymbol{D}_1})-\boldsymbol{X}(\boldsymbol{Z}^T\odot\boldsymbol{K}_{\boldsymbol{X}\boldsymbol{D}_2})\|_F\leq\frac{c}{\sigma\sqrt{n}}\|\boldsymbol{X}\|_2\|\boldsymbol{D}_1-\boldsymbol{D}_2\|_F$, *where c is a small constant.*

Proof. Since $\boldsymbol{Z}=\min_{\boldsymbol{Z}}\frac{1}{2}\|\phi(\boldsymbol{X})-\phi(\boldsymbol{D})\boldsymbol{Z}\|_F^2+\frac{\beta}{2}\|\boldsymbol{Z}\|_F^2$, we have

$$\boldsymbol{Z}=(\phi(\boldsymbol{D})^T\phi(\boldsymbol{D})+\beta\boldsymbol{I}_r)^{-1}\phi(\boldsymbol{D})^T\phi(\boldsymbol{X}).\tag{10}$$

Denote $\phi(\boldsymbol{D})=\boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T=\boldsymbol{U}\mathrm{diag}(\lambda_1,\cdots,\lambda_r)\boldsymbol{V}^T$ (the singular value decomposition), we have $\phi(\boldsymbol{X})=\boldsymbol{U}\hat{\boldsymbol{S}}\hat{\boldsymbol{V}}^T=\boldsymbol{U}\mathrm{diag}(\hat{\lambda}_1,\cdots,\hat{\lambda}_r)\hat{\boldsymbol{V}}^T$ because $\phi(\boldsymbol{X})$ and $\phi(\boldsymbol{D})$ have the same column basis. Then

$$\begin{aligned}\boldsymbol{Z}=&\boldsymbol{V}(\boldsymbol{S}^2+\beta\boldsymbol{I})^{-1}\boldsymbol{S}\hat{\boldsymbol{S}}\hat{\boldsymbol{V}}^T\\ =&\boldsymbol{V}\mathrm{diag}(\tfrac{\lambda_1\hat{\lambda}_1}{\lambda_1^2+\beta},\cdots,\tfrac{\lambda_r\hat{\lambda}_r}{\lambda_r^2+\beta})\hat{\boldsymbol{V}}^T.\end{aligned}\tag{11}$$

Suppose β is large enough, we have $\frac{\lambda_i\hat{\lambda}_i}{\lambda_i^2+\beta}<1$ for $i=1,\cdots,r$. It follows that $\|\boldsymbol{Z}\|_F^2<r$ and $E[z_{ij}^2]<\frac{1}{n}$, which indicates that

$$\begin{cases}\sigma_z=E[|z_{ij}-\mu_z|]<\frac{1}{\sqrt{n}},\\ -\frac{1}{\sqrt{n}}<\mu_z=E[z_{ij}]<\frac{1}{\sqrt{n}}.\end{cases}\tag{12}$$

According to Chebyshev's inequality, we have

$$\Pr(|z_{ij}| > \tfrac{c_0}{\sqrt{n}} + \tfrac{1}{\sqrt{n}}) < \tfrac{1}{c_0^2}. \tag{13}$$

Therefore, $|z_{ij}| < \frac{c_0}{\sqrt{n}}$ holds with high probability provided that c_0 is large enough. Suppose $z_{ij} \sim \mathcal{N}(\mu_z, \sigma_z^2)$, we have

$$\Pr(|z_{ij}| > \tfrac{c_0}{\sqrt{n}}) < e^{-0.5c_0^2} \tag{14}$$

according to the upper bound of Q-function of normal distribution. Then using union bound, we obtain

$$\Pr\big(|z_{ij}| < \tfrac{c_0}{\sqrt{n}}, \forall(i,j)\big) < 1 - nre^{-0.5c_0^2}, \tag{15}$$

It is equivalent to

$$\Pr\big(|z_{ij}| < \tfrac{c_1}{\sqrt{n}}, \forall(i,j)\big) < 1 - \tfrac{1}{(nr)^{c_0-1}}, \tag{16}$$

where $c_1 = \sqrt{2c_0\log(nr)}$.

On the other hand, the partial gradient of entry (i,j) of $\boldsymbol{K}_{\boldsymbol{XD}}$ in terms of $\boldsymbol{D}_{:j}$ (the j-th column of $\boldsymbol{D}$) can be given by

$$\tfrac{\partial \boldsymbol{K}_{\boldsymbol{XD}}(i,j)}{\partial \boldsymbol{D}_{:j}} = -\tfrac{1}{\sigma^2}(\boldsymbol{X}_{:i} - \boldsymbol{D}_{:j})\exp(-\tfrac{\|\boldsymbol{X}_{:i}-\boldsymbol{D}_{:j}\|^2}{2\sigma^2}). \tag{17}$$

Because $|x\exp(\frac{-x^2}{2\sigma^2})| \le \sigma\exp(-0.5) < 0.61\sigma$, we have $|\frac{\partial \boldsymbol{K}_{\boldsymbol{XD}}(i,j)}{\partial \boldsymbol{D}_{kj}}| < \frac{c_2}{\sigma}$ for some constant c_2. Then

$$\|\boldsymbol{K}_{\boldsymbol{XD}_1} - \boldsymbol{K}_{\boldsymbol{XD}_2}\|_F \le \tfrac{c_3}{\sigma}\|\boldsymbol{D}_1 - \boldsymbol{D}_2\|_F \tag{18}$$

some small constant c_3.

According to the above analysis, we get

$$\begin{aligned}
&\|\boldsymbol{X}(\boldsymbol{Z}^T \odot \boldsymbol{K}_{\boldsymbol{XD}_1}) - \boldsymbol{X}(\boldsymbol{Z}^T \odot \boldsymbol{K}_{\boldsymbol{XD}_2})\|_F \\
\le&\|\boldsymbol{X}\|_2\|\boldsymbol{Z}^T \odot (\boldsymbol{K}_{\boldsymbol{XD}_1} - \boldsymbol{K}_{\boldsymbol{XD}_2})\|_F \\
\le&\tfrac{c_1}{\sqrt{n}}\|\boldsymbol{X}\|_2\|\boldsymbol{K}_{\boldsymbol{XD}_1} - \boldsymbol{K}_{\boldsymbol{XD}_2}\|_F \\
\le&\tfrac{c_1}{\sqrt{n}}\tfrac{c_3}{\sigma}\|\boldsymbol{X}\|_2\|\boldsymbol{D}_1 - \boldsymbol{D}_2\|_F \\
=&\tfrac{c}{\sigma\sqrt{n}}\|\boldsymbol{X}\|_2\|\boldsymbol{D}_1 - \boldsymbol{D}_2\|_F.
\end{aligned} \tag{19}$$

□

1.3. Proof for Lemma 5

Lemma 5. *For sufficiently small η, Algorithm 1 converges to a stationary point.*

Proof. Denote the objective function of (7) by $\ell(\boldsymbol{Z}, \boldsymbol{D}, \boldsymbol{X})$, which is lower-bounded by at least 0. When $\eta = 0$, as the three subproblems are well addressed and do not diverge, we have $\ell(\boldsymbol{Z}_{t+1}, \boldsymbol{D}_{t+1}, \boldsymbol{X}_{t+1}) < \ell(\boldsymbol{Z}_{t+1}, \boldsymbol{D}_{t+1}, \boldsymbol{X}_t) < \ell(\boldsymbol{Z}_{t+1}, \boldsymbol{D}_t, \boldsymbol{X}_t) < \ell(\boldsymbol{Z}_t, \boldsymbol{D}_t, \boldsymbol{X}_t)$. It indicates that $\Delta_t = \ell(\boldsymbol{Z}_t, \boldsymbol{D}_t, \boldsymbol{X}_t) - \ell(\boldsymbol{Z}_{t+1}, \boldsymbol{D}_{t+1}, \boldsymbol{X}_{t+1}) \to 0$ when $t \to \infty$. When $\Delta_t = 0$, the gradient of $\ell(\boldsymbol{Z}_t, \boldsymbol{D}_t, \boldsymbol{X}_t)$ is 0. Then Algorithm 1 converges to a stationary point.

When $\eta > 0$ and take $\boldsymbol{D}$ as an example, because $\boldsymbol{\Delta}_{D,t}$ is not exact enough, we decompose $\boldsymbol{\Delta}_{D,t}$ as $\boldsymbol{\Delta}_{D,t} = c_t\boldsymbol{\Delta}^*_{D,t} + \boldsymbol{\Delta}'_{D,t}$, where $0 < c_t < 1$ and $\boldsymbol{\Delta}^*_{D,t}$ is nearly optimal at iteration t. Similarly, we have $\boldsymbol{\Delta}_{D,t-1} = c_{t-1}\boldsymbol{\Delta}^*_{D,t} + \boldsymbol{\Delta}'_{D,t-1}$. Then $\widehat{\boldsymbol{\Delta}}_t = (c_t + c_{t-1}\eta + \cdots + c_0\eta^t)\boldsymbol{\Delta}^*_{D,t} + \boldsymbol{\epsilon}'$, where $\boldsymbol{\epsilon}' = \sum_{i=0}^t \eta^i\boldsymbol{\Delta}'_{D,i}$. $\boldsymbol{\epsilon}'$ could be small compared to $\boldsymbol{\Delta}'_{D,t}$ because the signs of elements of $\boldsymbol{\Delta}'_{D,0}, \cdots, \boldsymbol{\Delta}'_{D,t}$ may change. Suppose c_t and η are small enough such that $c_t < c_t + c_{t-1}\eta + \cdots + c_0\eta^t < 1$, then $\widehat{\boldsymbol{\Delta}}_t$ is closer than $\boldsymbol{\Delta}_t$ to $\boldsymbol{\Delta}^*_t$. It indicates $\ell(\boldsymbol{Z}_{t+1}, \boldsymbol{D}_t - \widehat{\boldsymbol{\Delta}}_t, \boldsymbol{X}_t) < \ell(\boldsymbol{Z}_{t+1}, \boldsymbol{D}_t - \boldsymbol{\Delta}_t, \boldsymbol{X}_t) < \ell(\boldsymbol{Z}_{t+1}, \boldsymbol{D}_t, \boldsymbol{X}_t)$. That is why the momentum can accelerate the convergence.

More formally, take $\boldsymbol{D}$ with polynomial kernel as an example, in Lemma 3, we have proved $\ell(\boldsymbol{Z}, \boldsymbol{D} - \boldsymbol{\Delta}_D, \boldsymbol{X}) - \ell(\boldsymbol{Z}, \boldsymbol{D}, \boldsymbol{X}) \le -\frac{1}{2\tau}\mathrm{Tr}(\boldsymbol{g}_D\boldsymbol{H}_D^{-1}\boldsymbol{g}_D^T)$. As $\boldsymbol{\Delta}_D = \frac{1}{\tau}\boldsymbol{g}_D\boldsymbol{H}_D^{-1}$, we have

$$\ell(\boldsymbol{Z}, \boldsymbol{D} - \boldsymbol{\Delta}_D, \boldsymbol{X}) - \ell(\boldsymbol{Z}, \boldsymbol{D}, \boldsymbol{X}) \le -\tfrac{\tau}{2}\mathrm{Tr}(\boldsymbol{\Delta}_D\boldsymbol{H}\boldsymbol{\Delta}_D^T).$$

When momentum is used, $\boldsymbol{\Delta}_D$ is replaced by $\boldsymbol{\Delta}_D + \eta\widehat{\boldsymbol{\Delta}}_D$. Using the Taylor approximation similar to Lemma 3, we have

$$\begin{aligned}
&\ell(\boldsymbol{Z}, \boldsymbol{D} - \boldsymbol{\Delta}_D - \eta\widehat{\boldsymbol{\Delta}}_D, \boldsymbol{X}) \\
\le&\ell(\boldsymbol{Z}, \boldsymbol{D} - \boldsymbol{\Delta}_D, \boldsymbol{X}) + \langle \boldsymbol{G}_\eta, \eta\widehat{\boldsymbol{\Delta}}_D\rangle + \tfrac{\eta^2\tau}{2}\mathrm{Tr}(\widehat{\boldsymbol{\Delta}}_D\boldsymbol{H}\widehat{\boldsymbol{\Delta}}_D^T),
\end{aligned} \tag{20}$$

where $\boldsymbol{G}_\eta$ denotes the partial derivative of ℓ at $\boldsymbol{D} - \boldsymbol{\Delta}_D$. It follows that

$$\begin{aligned}
&\ell(\boldsymbol{Z}, \boldsymbol{D} - \boldsymbol{\Delta}_D - \eta\widehat{\boldsymbol{\Delta}}_D, \boldsymbol{X}) \\
\le&\ell(\boldsymbol{Z}, \boldsymbol{D} - \boldsymbol{\Delta}_D, \boldsymbol{X}) + \eta^2\tau\mathrm{Tr}(\widehat{\boldsymbol{\Delta}}_D\boldsymbol{H}\widehat{\boldsymbol{\Delta}}_D^T).
\end{aligned} \tag{21}$$

If $\eta\widehat{\boldsymbol{\Delta}}_D$ is a descent value, we have

$$\begin{aligned}
&\ell(\boldsymbol{Z}, \boldsymbol{D} - \boldsymbol{\Delta}_D - \eta\widehat{\boldsymbol{\Delta}}_D, \boldsymbol{X}) \\
<&\ell(\boldsymbol{Z}, \boldsymbol{D} - \boldsymbol{\Delta}_D, \boldsymbol{X}) \\
\le&\ell(\boldsymbol{Z}, \boldsymbol{D}, \boldsymbol{X}) - \tfrac{\tau}{2}\mathrm{Tr}(\boldsymbol{\Delta}_D\boldsymbol{H}\boldsymbol{\Delta}_D^T).
\end{aligned} \tag{22}$$

Otherwise, we have

$$\begin{aligned}
&\ell(\boldsymbol{Z}, \boldsymbol{D} - \boldsymbol{\Delta}_D - \eta\widehat{\boldsymbol{\Delta}}_D, \boldsymbol{X}) \\
\le&\ell(\boldsymbol{Z}, \boldsymbol{D}, \boldsymbol{X}) - \tfrac{\tau}{2}\mathrm{Tr}(\boldsymbol{\Delta}_D\boldsymbol{H}\boldsymbol{\Delta}_D^T) + \eta^2\tau\mathrm{Tr}(\widehat{\boldsymbol{\Delta}}_D\boldsymbol{H}\widehat{\boldsymbol{\Delta}}_D^T).
\end{aligned} \tag{23}$$

Since $\boldsymbol{H}$ is positive definite, we have

$$\ell(\boldsymbol{Z}, \boldsymbol{D} - \boldsymbol{\Delta}_D - \eta\widehat{\boldsymbol{\Delta}}_D, \boldsymbol{X}) \le \ell(\boldsymbol{Z}, \boldsymbol{D}, \boldsymbol{X}) \tag{24}$$

if η is small enough. Then similar to the case of $\eta = 0$, the convergence can be proved. □

1.4. Proof for Lemma 6

Lemma 6. *Updating $\boldsymbol{D}$ as $\boldsymbol{D} - \boldsymbol{\Delta}_D$ does not diverge and $\hat{\ell}(\boldsymbol{z}, [\boldsymbol{x}]_{\bar{\omega}}, \boldsymbol{D} - \boldsymbol{\Delta}_D) - \hat{\ell}(\boldsymbol{z}, [\boldsymbol{x}]_{\bar{\omega}}, \boldsymbol{D}) \le -\frac{1}{2\tau\tau_0}\|\nabla_D\hat{\ell}\|_F^2$ provided that $\tau > 1$, where $\tau_0 = \|\boldsymbol{z}\boldsymbol{z}^T \odot \boldsymbol{W}_2 + \alpha\boldsymbol{W}_2 \odot \boldsymbol{I}_r\|_2$.*

Proof. Fixing $\boldsymbol{W}_1$ and $\boldsymbol{W}_2$, we have $\|\nabla_{D_1}\hat{\ell} - \nabla_{D_2}\hat{\ell}\|_F \leq \|q\boldsymbol{z}\boldsymbol{z}^T \odot \boldsymbol{W}_2 + q\alpha\boldsymbol{W}_2 \odot \boldsymbol{I}_r\|_2\|\boldsymbol{D}_1 - \boldsymbol{D}_2\|_F$, which means the Lipschitz constant of $\hat{\ell}$'s gradient can be estimated as $\tau_0 = \|q\boldsymbol{z}\boldsymbol{z}^T \odot \boldsymbol{W}_2 + q\alpha\boldsymbol{W}_2 \odot \boldsymbol{I}_r\|_2$. It follows that

$$\begin{aligned}\hat{\ell}(\boldsymbol{z}, [\boldsymbol{x}]_{\bar{\omega}}, \boldsymbol{D}) \leq & \hat{\ell}(\boldsymbol{z}, [\boldsymbol{x}]_{\bar{\omega}}, \boldsymbol{D}_0) + \langle \nabla_{\boldsymbol{D}}\hat{\ell}, \boldsymbol{D} - \boldsymbol{D}_0\rangle \\ & + \tfrac{\tau\tau_0}{2}\|\boldsymbol{D} - \boldsymbol{D}_0\|_F^2,\end{aligned} \tag{25}$$

where $\tau > 1$. We minimize the right part of (25) and get

$$\boldsymbol{D} = \boldsymbol{D}_0 - \tfrac{1}{\tau\tau_0}\nabla_{\boldsymbol{x}}\hat{\ell} := \boldsymbol{D}_0 - \boldsymbol{\Delta}_{\boldsymbol{D}}. \tag{26}$$

Substituting (26) into (25), we have

$$\hat{\ell}(\boldsymbol{z}, [\boldsymbol{x}]_{\bar{\omega}}, \boldsymbol{D}_0 - \boldsymbol{\Delta}_{\boldsymbol{D}}) - \hat{\ell}(\boldsymbol{z}, [\boldsymbol{x}]_{\bar{\omega}}, \boldsymbol{D}_0) \leq -\tfrac{1}{2\tau\tau_0}\|\nabla_{\boldsymbol{D}}\hat{\ell}\|_F^2. \tag{27}$$

□

1.5. Derivation for (36)

As the number of observed entries in each column of $\boldsymbol{X}$ is $o_{\boldsymbol{X}} = \rho m$, the number of observed entries in each column of $\phi(\boldsymbol{X}) \in \mathbb{R}^{\bar{m}\times n}$ is

$$o_{\phi(\boldsymbol{x})} = \tbinom{\rho m+q}{q}, \tag{28}$$

where ϕ is a q-order polynomial map. It is known that the number of observed entries in $\phi(\boldsymbol{X})$ should be larger than the number of degrees of freedom of $\phi(\boldsymbol{X})$, otherwise it is impossible to determine $\phi(\boldsymbol{X})$ uniquely among all rank-r matrices of size $\bar{m} \times n$ [11]. Then we require

$$no_{\phi(\boldsymbol{x})} > nr + (\bar{m} - r)r, \tag{29}$$

where $\bar{m} = \binom{m+q}{q}$ and $r = \binom{d+pq}{pq}$. Substituting (28) into (29) and dividing both sides with $\bar{m}$, we get

$$\frac{\binom{\rho m+q}{q}}{\binom{m+q}{q}} > \frac{nr + (\bar{m} - r)r}{n\bar{m}}. \tag{30}$$

Since

$$\frac{\binom{\rho m+q}{q}}{\binom{m+q}{q}} = \frac{(\rho m+q)(\rho m+q-1)\cdots(\rho m+1)}{(m+q)(m+q-1)\cdots(m+1)}, \tag{31}$$

we have $\binom{\rho m+q}{q}/\binom{m+q}{q} \approx \rho^q$ for small q. It follows that

$$\begin{aligned}\rho > & \left(\tfrac{nr+(\bar{m}-r)r}{n\bar{m}}\right)^{\frac{1}{q}} = \left(\tfrac{r}{n} + \tfrac{r}{\bar{m}} - \tfrac{r^2}{n\bar{m}}\right)^{\frac{1}{q}} \\ = & \left(\tfrac{u\binom{d+pq}{pq}}{n} + \tfrac{u\binom{d+pq}{pq}}{\binom{m+q}{q}} - \tfrac{u^2\binom{d+pq}{pq}^2}{n\binom{m+q}{q}}\right)^{\frac{1}{q}} \\ := & \kappa(m, n, d, p, q, u).\end{aligned} \tag{32}$$

1.6. Derivation for (37)

We reformulate RBF kernel as

$$\begin{aligned}k(\boldsymbol{x}, \boldsymbol{y}) = & \exp\left(-\tfrac{1}{2\sigma^2}(\|\boldsymbol{x}\|^2 + \|\boldsymbol{y}\|^2)\right)\exp\left(\tfrac{1}{\sigma^2}\langle \boldsymbol{x}, \boldsymbol{y}\rangle\right) \\ := & C\sum_{k=0}^{\infty}\frac{\langle \boldsymbol{x}, \boldsymbol{y}\rangle^k}{\sigma^{2k}k!} \\ = & C\sum_{k=0}^{q}\frac{\langle \boldsymbol{x}, \boldsymbol{y}\rangle^k}{\sigma^{2k}k!} + O(\frac{c^{q+1}}{(q+1)!}),\end{aligned} \tag{33}$$

where $0 < c < 1$ provided that $\sigma^2 > |\boldsymbol{x}^T\boldsymbol{y}|$ and $C = \exp\left(-\frac{1}{2\sigma^2}(\|\boldsymbol{x}\|^2 + \|\boldsymbol{y}\|^2)\right)$. We see that RBF kernel can be approximated by a weighted sum of polynomial kernels with orders $0, 1, \cdots, q$, where the error is $O(\frac{c^{q+1}}{(q+1)!})$. The feature map of the weighted sum is a q-order polynomial map, denoted by $\hat{\phi}$. Then it follows from (33) that

$$\phi(\boldsymbol{x})^T\phi(\boldsymbol{x}) = \hat{\phi}(\boldsymbol{x})^T\hat{\phi}(\boldsymbol{x}) + O(\tfrac{c^{q+1}}{(q+1)!}), \tag{34}$$

and further

$$\phi_i(\boldsymbol{x}) = \hat{\phi}_i(\boldsymbol{x}) + O(\sqrt{\tfrac{c^{q+1}}{(q+1)!}}), \tag{35}$$

in which we have assumed that the signs of $\phi_i(\boldsymbol{x})$ and $\hat{\phi}_i(\boldsymbol{x})$ are the same because it has no influence on the feature map. It means the feature map ϕ of RBF kernel can be well approximated by a q-order polynomial map, where the approximation error is $O(\sqrt{\frac{c^{q+1}}{(q+1)!}})$ and could be nearly zero. Therefore, $\rho > \kappa(m, n, d, p, q, u)$ in (32) holds for RBF kernel with error $O(\sqrt{\frac{c^{q+1}}{(q+1)!}})$ in recovering $\phi(\boldsymbol{X})$. When $\phi(\boldsymbol{X})$ is recovered, $\boldsymbol{X}$ is naturally recovered because $\boldsymbol{X}$ itself is the first-order feature in $\phi(\boldsymbol{X})$.

2. More about the experiments

2.1. An intuitive example

We use a simple example of nonlinear data to intuitively show the performance of our high-rank matrix completion method KFMC. Specifically, we sample 100 data points from the following twisted cubic function

$$x_1 = s,\ x_2 = s^2,\ x_3 = s^3, \tag{36}$$

where $s \sim \mathcal{U}(-1, 1)$. Then we obtain a 3×100 matrix, which is of full-rank. For each data point (column), we randomly remove one entry. The recovery results of low-rank matrix completion and our KFMC are shown in Figure 1. We see that LRMC absolutely failed because it cannot handle full-rank matrix. On the contrary, our KFMC recovered the missing entries successfully. The performance of KFMC at different iteration is shown in Figure 2, which demonstrated that KFMC shaped the data into the curve gradually. It is worth mentioning that when we remove two entries of each column of the matrix, KFMC cannot recovery the missing entries because the number of observed entries is smaller than the latent dimension of the data.

2.2. Compared methods and parameter settings

For offline matrix completion, our KFMC with polynomial kernel and KFMC with RBF kernel are compared with the following methods.

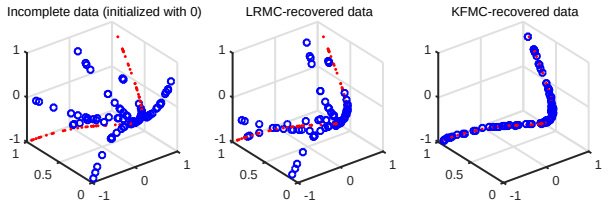

Figure 1: Recovery result on data drawn from (36) (the red points are the complete data)

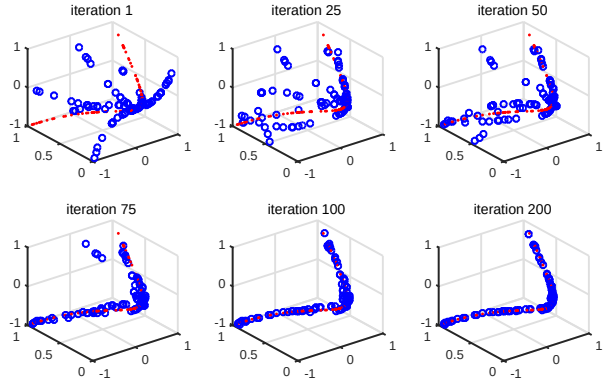

Figure 2: KFMC recovery performance at different iteration on data drawn from (36)

LRF (low-rank factorization based matrix completion [12]). LRF is solved by alternating minimization. The matrix rank and regularization parameter are searched within a broad range such that the best results are reported in our experiments.

NNM (nuclear norm minimization [2]). NNM is solved by inexact augmented lagrange multiplier [9] and has no parameter to determine beforehand. Therefore it is good baseline for our evaluation.

VMC (algebraic variety model for matrix completion [11]). In VMC, second-order polynomial kernel is used, where the hyper-parameter is chosen from $\{1, 10, 100\}$. The parameter of Schatten-p norm is set as 0.5, which often performs the best. To reduce its computational cost, randomized SVD [7], instead of full SVD, is performed.

NLMC (nonlinear matrix completion [5]). In NLMC, RBF kernel is used. The parameter σ of the kernel is chosen from $\{0.5\bar{d}, 1\bar{d}, 3\bar{d}\}$, where $\bar{d}$ is the average distance of all pair-wise data points. The parameter of Schatten-p norm is set as 0.5 and randomized SVD is also performed.

In our KFMC(Poly) method, second order polynomial kernel is used, where the hyper-parameter is set as 1. The regularization parameters α and β are chosen from $\{0.01, 0.1\}$.

In our KFMC(RBF), the setting of parameter σ is the same as that of NLMC. The regularization parameter β is chosen from $\{0.001, 0.0001\}$ while α does not matter. The parameter r of KFMC(Poly) and KFMC(RBF) are chosen from $\{0.5m, 1m, 2m\}$, where m is the row dimension of the matrix.

For online matrix completion, the parameter setting of OL-KFMC is similar to that of KFMC. Our OL-KFMC(Poly) and OL-KFMC(RBF) are compared with the following methods.

GROUSE [1][1]. The learning rate and matrix rank are searched within large ranges to provide the best performances.

OL-DLSR (online dictionary learning and sparse representation based matrix completion). OL-DLSR is achieved by integrating [10] with [6]. It solves the following problem

$$\underset{\boldsymbol{D}\in\mathcal{C},\boldsymbol{z}}{\text{minimize}} \frac{1}{2}\|\boldsymbol{\omega}\odot(\boldsymbol{x}-\boldsymbol{D}\boldsymbol{z})\|^2+\lambda\|\boldsymbol{z}\|_1 \qquad (37)$$

for a set of incomplete data columns $\{\boldsymbol{x}\}$. $\boldsymbol{\omega}$ is a binary vector with $\omega_i = 0$ if entry i of $\boldsymbol{x}$ is missing and $\omega_i = 1$ otherwise. According to [6], the method can recover high-rank matrices online when the data are drawn from a union of subspaces. We determine λ and the number of columns of $\boldsymbol{D}$ carefully to give the best performances of OL-DLSR in our experiments.

OL-LRF (online LRF [12, 8]). OL-LRF is similar to OL-DLSR. The only difference is that $\|\boldsymbol{z}\|_1$ is replaced by $\frac{1}{2}\|\boldsymbol{z}\|_F^2$. In OL-LRF, the matrix rank is carefully determined to give the best performances in our experiments.

For out-of-sample extension of matrix completion, our OSE-KFMC is compared with the following two methods.

OSE-LRF First, perform SVD on a complete training data matrix, i.e., $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T$, where $\boldsymbol{U} \in \mathbb{R}^{m\times r}$, $\boldsymbol{S} \in \mathbb{R}^{r\times r}$, $\boldsymbol{V} \in \mathbb{R}^{n\times r}$, and $r = \text{rank}(\boldsymbol{X})$. For a new incomplete data column $\boldsymbol{x}$, the missing entries are recovered as

$$\boldsymbol{x}_{\bar{\omega}} = \boldsymbol{U}_{\bar{\omega}}(\boldsymbol{U}_\omega^T\boldsymbol{U}_\omega + \lambda\boldsymbol{I}_{|\omega|})^{-1}\boldsymbol{U}_\omega^T\boldsymbol{x}_\omega, \qquad (38)$$

where ω denotes the locations of observed entries, $\bar{\omega}$ denotes the locations of missing entries, λ is a small constant, and $\boldsymbol{U}_{\bar{\omega}}$ consists of the rows of $\boldsymbol{U}$ corresponding to $\bar{\omega}$.

[1] http://web.eecs.umich.edu/∼girasole/?p=110

OSE-DLSR First, a dictionary $\boldsymbol{D}$ is learned by the method of [10] from the training data. Given a new incomplete data $\boldsymbol{x}$, we can obtain the sparse coefficient as

$$\boldsymbol{z} = \min_{\boldsymbol{z}} \tfrac{1}{2}\|\boldsymbol{\omega} \odot (\boldsymbol{x} - \boldsymbol{D}\boldsymbol{z})\|^2 + \lambda\|\boldsymbol{z}\|_1. \tag{39}$$

Finally, the missing entries of $\boldsymbol{x}$ can be recovered as $\boldsymbol{x}_{\bar{\omega}} = \boldsymbol{D}_{\bar{\omega}}\boldsymbol{z}$.

The experiments are conducted with MATLAB on a computer with Inter-i7-3.4GHz Core and 16 GB RAM. The maximum iteration of each offline matrix completion method is 500, which is often enough to converge or give a high recovery accuracy. It also provides a baseline to compare the computational costs of VMC, NLMC, and KFMC.

2.3. Synthetic data

Take the case of three nonlinear subspaces as an example, the optimization curves of our KFMC with different momentum parameter η are shown in Figure 3. We see that a larger η can lead to a faster convergence. Particularly, compared with KFMC(Poly), KFMC(RBF) requires fewer iterations to converge, while in each iteration the computational cost of the former is a little bit higher than that of the latter. In this paper, we set $\eta = 0.5$ for all experiments.

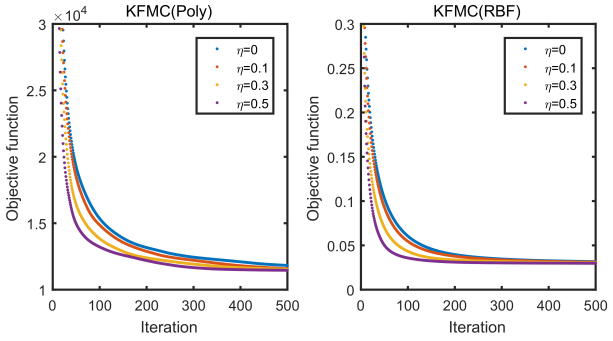

Figure 3: Optimization of KFMC with different momentum parameter η

Figure 4 shows the online KFMC's iterative changes of empirical cost function

$$g_t(\boldsymbol{D}) := \frac{1}{t}\sum_{j=1}^{t} \ell([\boldsymbol{x}_j]_{\omega_j}, \boldsymbol{D}) \tag{40}$$

and empirical recovery error

$$e_t(\boldsymbol{x}) := \frac{1}{t}\sum_{j=1}^{t} \frac{\|\boldsymbol{x}_j - \hat{\boldsymbol{x}}_j\|}{\|\boldsymbol{x}_j\|}, \tag{41}$$

where $\hat{\boldsymbol{x}}_j$ denotes the recovered column and t is the number of online samples. At the beginning of the online learning (t is small), the recover errors and the values of cost function are high. With the increasing of t, the recover errors and the values of cost function decreased. In practice, we can re-pass the data to reduce the recovery errors. In addition, when t is large enough and the structure of the data is assumed to be fixed, we do not need to update $\boldsymbol{D}$. If the data structure changes according to time, we can just update $\boldsymbol{D}$ all the time in order to adapt to the changes.

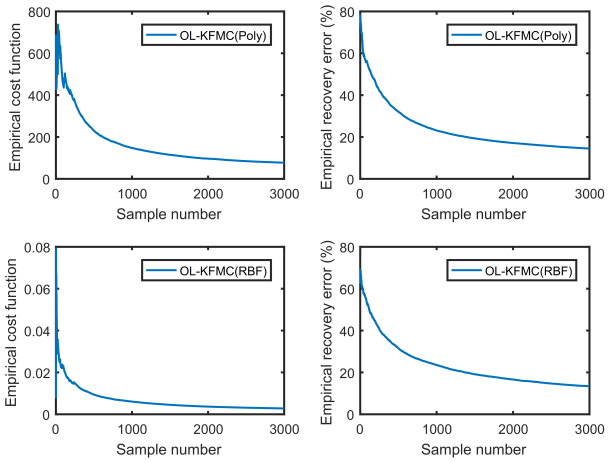

Figure 4: Empirical cost function and recovery error of online KFMC

In our experiments of online matrix completion, the reported recovery errors are the results after the data matrix was passed for 10 times. Figure 6 shows the matrix completion errors of different number of passes. Our OL-KFMC(Poly) and OL-KFMC(RBF) have the lowest recovery errors. The recovery errors of OL-LRF and GROUSE are considerably high because they are low-rank methods but the matrix in the experiment is of high-rank.

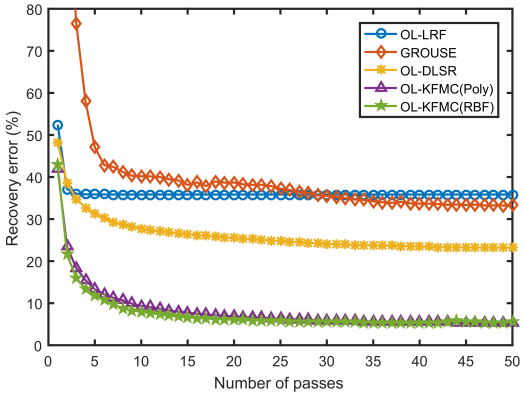

Figure 5: Matrix completion errors of different passes

2.4. Real data

For the experiments of subspace clustering on incomplete data of Hopkins 155 datasets [13], similar to [11], we conducted the following procedures. First, the two subsets of video sequences, *1R2RC* and *1R2TCR*, were uniformly downsampled to 6 frames. Then the sizes of the resulted data matrices are 12×459 and 12×556. Second, we randomly removed a fraction ($10\% \sim 70\%$) of the entries of the two matrices and then perform matrix completion to recover the missing entries. Finally, SSC (sparse subspace clustering [4]) were performed to segment the data into different clusters. For fair comparison, the parameter λ in SSC were chosen from $\{1, 10, 100\}$ and the best results were reported.

For the CMU motion capture data, similar to [3, 11], we use the trial $\#6$ of subject $\#56$ of the dataset, which is available at *http://mocap.cs.cmu.edu/*. The data consists of the following motions: throw punches, grab, skip, yawn, stretch, leap, lift open window, walk, and jump/bound. The data size is 62×6784. The data of each motion lie in a low-rank subspace and the whole data matrix is of full-rank [3]. To reduce the computational cost and increase the recovery difficulty, we sub-sampled the data to 62×3392. We considered two types of missing data pattern. The first one is randomly missing, for which we randomly removed 10% to 70% entries of the matrix. The second one is continuously missing, which is more practical and challenging. Specifically, for each row of the matrix, the missing entries were divided into 50 missing sequences, where the sequences are uniformly distributed and the length of each sequence is about 68δ. Here δ denotes the missing rate. The two missing data patterns are shown in Figure 6, in which the black pixel or region denotes the missing entries. For the online recovery, the number of passes for OL-LRF, GROUSE, OL-DLSR, and OL-KFMC are 10, 50, 10, and 5 respectively. The reason for this setting is that GROUSE requires large number of passes while the other methods especially our OL-KFMC requires fewer passes.

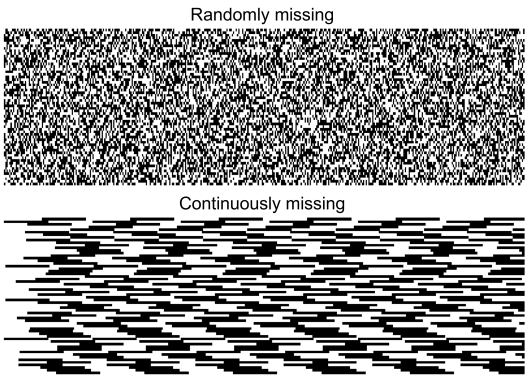

Figure 6: Two missing data patterns for motion capture data

References

[1] Laura Balzano, Robert Nowak, and Benjamin Recht. Online identification and tracking of subspaces from highly incomplete information. In *Communication, Control, and Computing (Allerton), 2010 48th Annual Allerton Conference on*, pages 704–711. IEEE, 2010. 4

[2] Emmanuel J. Candès and Benjamin Recht. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 9(6):717–772, 2009. 4

[3] Ehsan Elhamifar. High-rank matrix completion and clustering under self-expressive models. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 73–81. Curran Associates, Inc., 2016. 6

[4] E. Elhamifar and R. Vidal. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(11):2765–2781, 2013. 6

[5] Jicong Fan and Tommy W.S. Chow. Non-linear matrix completion. *Pattern Recognition*, 77:378 – 394, 2018. 4

[6] J. Fan, M. Zhao, and T. W. S. Chow. Matrix completion via sparse factorization solved by accelerated proximal alternating linearized minimization. *IEEE Transactions on Big Data*, pages 1–1, 2018. 4

[7] N. Halko, P. G. Martinsson, and J. A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 53(2):217–288, 2011. 4

[8] Chi Jin, Sham M Kakade, and Praneeth Netrapalli. Provable efficient online matrix completion via non-convex stochastic gradient descent. In *Advances in Neural Information Processing Systems*, pages 4520–4528, 2016. 4

[9] Zhouchen Lin, Minming Chen, and Yi Ma. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. *arXiv:1009.5055v3 [math.OC]*, 2010. 4

[10] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online dictionary learning for sparse coding. In *Proceedings of the 26th annual international conference on machine learning*, pages 689–696. ACM, 2009. 4, 5

[11] Greg Ongie, Rebecca Willett, Robert D. Nowak, and Laura Balzano. Algebraic variety models for high-rank matrix completion. In *Proceedings of the 34th International Conference on Machine Learning*, pages 2691–2700. PMLR, 2017. 3, 4, 6

[12] Ruoyu Sun and Zhi-Quan Luo. Guaranteed matrix completion via non-convex factorization. *IEEE Transactions on Information Theory*, 62(11):6535–6579, 2016. 4

[13] R. Tron and R. Vidal. A benchmark for the comparison of 3-d motion segmentation algorithms. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, June 2007. 6