Low-rank Matrix Completion: Guest Lecture for 551

Greg Ongie, University of Michigan

November 7, 2017

1 Optimization Formulation

- Let $\mathbf{Y} = [Y_{i,j}] \in \mathbb{R}^{m\times n}$. Suppose we observe $Y_{i,j}$ for all $(i,j) \in \Omega$ where $\Omega \subset \{1,...,m\} \times \{1,...,n\}$ is an index set of the obsered locations of size s.

- **Notation:** Define the linear *projection operator* $\mathcal{P}_\Omega : \mathbb{R}^{m\times n} \to \mathbb{R}^s$ by

$$\mathcal{P}_\Omega(\mathbf{X}) = [X_{i,j}]_{(i,j)\in\Omega} \quad \text{(vector of entries in } \Omega)$$

2×2 **matrix example:**

$$\mathbf{Y} = \begin{bmatrix} 2 & 5 \\ 6 & 7 \end{bmatrix}, \quad \Omega = \{(1,1),(1,2),(2,2)\} \simeq \underbrace{\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}}_{\text{“sampling mask”}}, \quad \mathcal{P}_\Omega(\mathbf{Y}) = \begin{bmatrix} 2 \\ 5 \\ 7 \end{bmatrix}$$

Allows us to write observations constraints compactly:

$$\mathcal{P}_\Omega(\mathbf{X}) = \mathcal{P}_\Omega(\mathbf{Y})$$

if and only if

$$X_{i,j} = Y_{i,j} \;\text{ for all }\; (i,j) \in \Omega.$$

- **Low-rank matrix completion problem:**
Find a matrix $\widehat{\mathbf{X}}$ such that
(1) $\widehat{\mathbf{X}}$ is low-rank
(2) $\mathcal{P}_\Omega(\widehat{\mathbf{X}}) = \mathcal{P}_\Omega(\mathbf{Y})$

- “Ideal” Optimization Formulation:

$$\hat{\mathbf{X}} = \arg\min_{\mathbf{X}} \; \mathrm{rank}(\mathbf{X}) \;\text{ subject to }\; \mathcal{P}_\Omega(\mathbf{X}) = \mathcal{P}_\Omega(\mathbf{Y}). \tag{rank-min}$$

In words, *find the matrix of minimum rank that agrees with the observed entries.*

- Challenges this approach:
  - Rank functional is non-convex.
  - No fast algorithm to solve this problem (“NP-hard”).

- In practice, often “noise” in samples: $\mathcal{P}_\Omega(\mathbf{X}) \approx \mathcal{P}_\Omega(\mathbf{X}_0)$.

- Solution is to “relax” the problem: Replace rank($\mathbf{X}$) with nuclear norm, and include data-fit term

$$\widehat{\mathbf{X}} = \arg\min_{\mathbf{X}} \underbrace{\|\mathcal{P}_\Omega(\mathbf{X}) - \mathcal{P}_\Omega(\mathbf{Y})\|^2}_{\text{“data fit”}} + \underbrace{\beta\|\mathbf{X}\|_*}_{\text{“regularizer”}} \tag{NN-min}$$

- **Recall:** From Nov. 2 class, closely related matrix denoising problem:

$$\widehat{\mathbf{X}} = \arg\min_{\mathbf{X}} \underbrace{\frac{1}{2}\|\mathbf{X} - \mathbf{Y}\|_F^2}_{\text{“data fit”}} + \underbrace{\beta\|\mathbf{X}\|_*}_{\text{“regularizer”}} \tag{SVST}$$

which has closed form solution

$$\widehat{\mathbf{X}} = SVST(\mathbf{Y}, \beta) = \sum_{k=1}^{r} [\sigma_k - \beta]_+ \mathbf{u}_k \mathbf{v}_k'$$

where $\mathbf{u}_1, ..., \mathbf{u}_r$ and $\mathbf{v}_1, ..., \mathbf{v}_r$ are the left and right singular vectors of $\mathbf{Y}$. Only difference with (NN-min) is the linear operator $\mathcal{P}_\Omega$ inside Frobenius norm. This penalizes data-fit only on observed set.

2 Iterative Soft-Thresholding Algorithm (ISTA)

- We will derive an efficient algorithm to solve (NN-min) that combines gradient descent with singular value soft-thresholding.

- **Note:** Gradient descent can also be applied to matrix functions $f : \mathbb{R}^{n\times m} \to \mathbb{R}$. We say $f(\mathbf{X})$ is smooth if all partial derivatives $\partial f(\mathbf{X})/\partial X_{i,j}$ exist, and we write $\nabla f(\mathbf{X})$ for the matrix of partial derivatives. I will move back and forth between matrix and vector functions and their gradients with the understanding that they are equivalent up to “reshaping”.

- **Recall:** If $f : \mathbb{R}^n \to \mathbb{R}$ is a smooth function we can solve

$$\min_{\mathbf{x}} f(\mathbf{x})$$

by *gradient descent*: initialize with $\mathbf{x}_0 \in \mathbb{R}^{m\times n}$ and for all $k = 0, 1, ...$ iterate

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha \nabla f(\mathbf{x}_k)$$

Typically the choice of α depends on the *Lipschitz constant* L of $\nabla f(\mathbf{x})$.
Example: quadratic f

$$f(\mathbf{x}) = \|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2$$

has Lipschitz gradient $L = 2\|\mathbf{A}'\mathbf{A}\|_2$, and $\alpha \in (0, 1/\|\mathbf{A}'\mathbf{A}\|_2)$ ensures convergence of gradient descent to the global optimum.

- We will extend gradient descent to solve problems of the type

$$\min_{\mathbf{x}} \underbrace{f(\mathbf{x})}_{\text{smooth}} + \underbrace{g(\mathbf{x})}_{\text{non-smooth}}$$

Why? Because (NN-min) has this form:

$$\min_{\mathbf{X}} \underbrace{\|\mathcal{P}_\Omega(\mathbf{X}) - \mathcal{P}_\Omega(\mathbf{X})\|_F^2}_{\text{smooth (quadratic)}} + \underbrace{\beta\|\mathbf{X}\|_*}_{\text{non-smooth}}$$

- **Key idea:** The gradient descent step $\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha\nabla f(\mathbf{x}_k)$ is equivalent to

$$\mathbf{x}_{k+1} = \arg\min_{x} \left\{ \underbrace{f(\mathbf{x}_k) + \langle \nabla f(\mathbf{x}), \mathbf{x} - \mathbf{x}_k\rangle}_{\text{first-order Taylor expansion}} + \frac{1}{2\alpha}\|\mathbf{x} - \mathbf{x}_k\|^2 \right\}$$

- **Interpretation:** we are "majorizing" the function $f(\mathbf{x})$ with quadratic surrogate function, and minimizing the surrogate function at each iteration. [Picture]

Proof. Removing terms that do not depend on $\mathbf{x}$ we have

$$\begin{aligned}
\mathbf{x}_{k+1} &= \arg\min_{\mathbf{x}} \left\{ \langle \nabla f(\mathbf{x}_k), \mathbf{x}\rangle + \frac{1}{2\alpha}\|\mathbf{x} - \mathbf{x}_k\|^2 \right\} \\
&= \arg\min_{\mathbf{x}} \left\{ \langle \nabla f(\mathbf{x}_k), \mathbf{x}\rangle + \frac{1}{2\alpha}\|\mathbf{x}\|^2 - \frac{1}{\alpha}\langle \mathbf{x}, \mathbf{x}_k\rangle + \frac{1}{2\alpha}\|\mathbf{x}_k\|^2 \right\} \\
&= \arg\min_{\mathbf{x}} \left\{ \frac{1}{2\alpha}\|\mathbf{x}\|^2 - \frac{1}{\alpha}\langle \mathbf{x}, \mathbf{x}_k - \alpha\nabla f(\mathbf{x}_k)\rangle \right\} \\
&= \arg\min_{\mathbf{x}} \left\{ \frac{1}{2\alpha}\|\mathbf{x} - (\mathbf{x}_k - \alpha\nabla f(\mathbf{x}_k))\|^2 \right\}.
\end{aligned}$$

The minimum happens where the objective is 0, hence $\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha\nabla f(\mathbf{x}_k)$. □

- Extend this new interpretation to minimize sum of smooth and non-smooth term: solve

$$\min_{\mathbf{x}} f(\mathbf{x}) + g(\mathbf{x})$$

by iterating

$$\begin{aligned}
\mathbf{x}_{k+1} &= \arg\min_{x} \left\{ f(\mathbf{x}_k) + \langle \nabla f(\mathbf{x}), \mathbf{x} - \mathbf{x}_k\rangle + \frac{1}{2\alpha}\|\mathbf{x} - \mathbf{x}_k\|^2 + g(\mathbf{x}) \right\} \\
&= \arg\min_{x} \left\{ \frac{1}{2\alpha}\|\mathbf{x} - (\mathbf{x}_k - \alpha\nabla f(\mathbf{x}))\|^2 + g(\mathbf{x}) \right\}.
\end{aligned}$$

Or, put compactly,

$$\begin{aligned}
\mathbf{y}_k &= \mathbf{x}_k - \alpha\nabla f(\mathbf{x}_k) \\
\mathbf{x}_{k+1} &= \arg\min_{\mathbf{x}} \left\{ \frac{1}{2\alpha}\|\mathbf{x} - \mathbf{y}_k\|^2 + g(\mathbf{x}) \right\}.
\end{aligned}$$

- Algorithm also guaranteed to converge to global minimum (for convex objectives) under similar conditions on α as for gradient descent. Typically choose $\alpha = \frac{1}{L}$, where L is the Lipschitz constant of $\nabla f(\mathbf{x})$.

- Now apply this to matrix completion problem (NN-min):
  $f(\mathbf{X}) = \frac{1}{2}\|\mathcal{P}_\Omega(\mathbf{X}) - \mathcal{P}_\Omega(\mathbf{X}_0)\|^2$ and $g(X) = \beta\,\|\mathbf{X}\|_*$,

  – **step-size** α: We can re-write f as quadratic in the variable $x = \mathrm{vec}(\mathbf{X})$, the vectorized matrix, as $\tilde{f}(\mathbf{x}) = \frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2$ where $\mathbf{A}$ contains rows of the $mn \times mn$ identity matrix, and $\mathbf{A}'\mathbf{A} = diag(\mathbf{1}_\Omega)$ where $\mathbf{1}_\Omega[i,j] = 1$ if $(i,j) \in \Omega$ and 0 elsewhere. Therefore $L = \|\mathbf{A}'\mathbf{A}\| = 1$, and we can choose the step-size $\alpha = \frac{1}{L} = 1$.

  – $\mathbf{Y}_k$ **update**: Again, let $\mathbf{x}_k = \mathrm{vec}(\mathbf{X}_k)$, then

  $$\nabla\tilde{f}(\mathbf{x}_k) = \mathbf{A}'(\mathbf{A}\mathbf{x}_k - \mathbf{b}) = \mathbf{A}'\mathbf{A}\mathbf{x}_k - \mathbf{A}'\mathbf{b}$$

  Re-writing this in a matrix variable $\mathbf{X}_k$, we have:

  $$\nabla f(\mathbf{X}_k) = \mathcal{P}_\Omega^*\mathcal{P}_\Omega(\mathbf{X}_k) - \mathcal{P}_\Omega^*\mathcal{P}_\Omega(\mathbf{Y})$$

  where $\mathcal{P}_\Omega^* : \mathbb{R}^K \to \mathbb{R}^{m\times n}$ maps a vector of samples to the matrix with those samples at locations Ω and zeros elsewhere. Hence we have

  $$\mathbf{Y}_k = \mathbf{X}_k - \alpha\nabla f(\mathbf{X}_k) = [\mathbf{X}_k - \mathcal{P}_\Omega^*\mathcal{P}_\Omega(\mathbf{X}_k)] + \mathcal{P}_\Omega^*\mathcal{P}_\Omega(\mathbf{Y}).$$

  We can also write this as:

  $$[\mathbf{Y}_k]_{i,j} = \begin{cases} [\mathbf{X}_k]_{i,j} & \text{if } (i,j) \notin \Omega \\ [\mathbf{Y}]_{i,j} & \text{if } (i,j) \in \Omega \end{cases},$$

  i.e., we set the entries of $\mathbf{Y}_k$ equal to the entries of $\mathbf{Y}$ on the observation set Ω, and equal the entries of $\mathbf{X}_k$ elsewhere.

  – $\mathbf{X}_{k+1}$ **update**: This becomes

  $$\mathbf{X}_{k+1} = \arg\min_{\mathbf{X}} \frac{1}{2}\|\mathbf{X} - \mathbf{Y}_k\|_F^2 + \beta\,\|\mathbf{X}\|_*$$

  Exactly soft-thresholding of singular values! Easy to implement.

  – Final Iterative Soft-thresholding Algorithm (ISTA) for (NN-min):
  initialize $\mathbf{X}_0 \in \mathbb{R}^{m\times n}$ and for all $k = 0, 1, 2, ...$ iterate

  $$\begin{aligned} \mathbf{Y}_k &= \mathbf{X}_k \\ \mathcal{P}_\Omega(\mathbf{Y}_k) &\leftarrow \mathcal{P}_\Omega(\mathbf{Y}) \quad \text{(put in known samples)} \\ \mathbf{X}_{k+1} &= SVST(\mathbf{Y}_k, \beta) \end{aligned}$$

3 Fast Iterative Soft-Thresholding Algorithm (FISTA)

- Modification of ISTA to allow for Nesterov acceleration:
  FISTA: Set $t_0 = 1$, and for all $k = 0, 1, 2, ...$ iterate

$$\begin{aligned}
\widehat{\mathbf{Y}}_k &= \mathbf{Y}_k \\
\mathcal{P}_\Omega(\widehat{\mathbf{Y}}_k) &\leftarrow \mathcal{P}_\Omega(\mathbf{Y}) \quad \text{(put in known samples)} \\
\mathbf{X}_{k+1} &= SVST(\widehat{\mathbf{Y}}_k, \beta) \\
t_{k+1} &= \frac{1 + \sqrt{1 + 4t_k^2}}{2} \\
\mathbf{Y}_{k+1} &= \mathbf{X}_k + \frac{t_k - 1}{t_{k+1}}(\mathbf{X}_k - \mathbf{X}_{k+1})
\end{aligned}$$