Low-rank Approximations for Incomplete Matrices

Liliya Ageeva
Sergey Makarychev
Aleksandr Rozhnov
Anton Zhevnerchuk

Mentor: Maksim Rakhuba

Background

There are some challenging industrial problems in which only incomplete data is available and the goal is to “complete” the data.

Background

There are some challenging industrial problems in which only incomplete data is available and the goal is to "complete" the data.

- Recommender systems

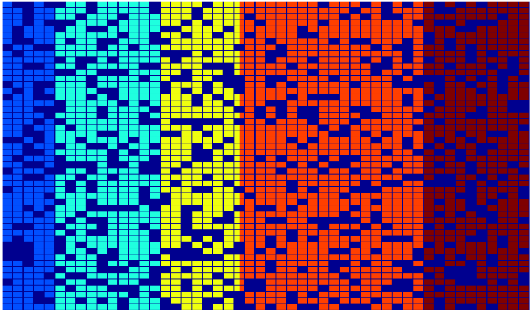
Background

There are some challenging industrial problems in which only incomplete data is available and the goal is to "complete" the data.

- Recommender systems
- Repairing damaged files

There are some challenging industrial problems in which only incomplete data is available and the goal is to "complete" the data.

- Recommender systems
- Repairing damaged files



Problem Formulation

Skoltech

General Formulation

Input:

- Data dimensions (M, N)
- Set of known entries Ω and values at them (sparse matrix X)

Output:

- Low-rank approximation Z

Problem Formulation

Skoltech

General Formulation

Input:

- Data dimensions (M, N)
- Set of known entries Ω and values at them (sparse matrix X)

Output:

- Low-rank approximation Z

Approach I

Fix rank, minimize error at the known entries.

Approach II

Set maximum acceptable error at the known entries, minimize rank.

Implemented approaches

- Alternating Least Squares (Approach I)
- Riemannian optimization [Vandereycken, 2012] (Approach I)
- Soft-Input [Mazumder, Hastie, Tibshirani, 2010] (Approach II)

Alternating Least Squares(ALS)

Skoltech

$$\underset{Z}{\text{minimize}} \quad \frac{1}{2} \sum_{(i,j)\in\Omega} (X_{ij} - Z_{ij})^2 \quad \text{s.t.} \quad \mathrm{rank}(Z) = K$$

Alternating Least Squares (ALS)

- Find Z in the form $Z = U^T V$, $U \in \mathbb{R}^{K\times M}$, $V \in \mathbb{R}^{K\times N}$
- Update U and V independently until convergence
- At each step optimal U and V can be found analytically

$$\underset{Z}{\text{minimize}} \quad \frac{1}{2}\sum_{(i,j)\in\Omega}(X_{ij}-Z_{ij})^2 \quad \text{s.t.} \quad \mathrm{rank}(Z)=K$$

Alternating Least Squares (ALS)

- Find Z in the form $Z = U^T V$, $U \in \mathbb{R}^{K\times M}$, $V \in \mathbb{R}^{K\times N}$
- Update U and V independently until convergence
- At each step optimal U and V can be found analytically

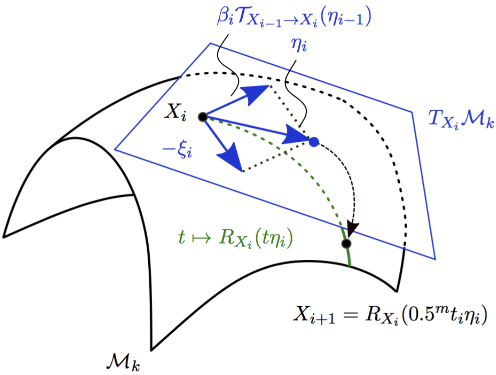
Avoiding overfitting: add regularization term $\lambda(\|U\|_F^2 + \|V\|_F^2)$, still explicit formulas for optimal U and V^T at each step.

Riemannian optimization

- Matrices of fixed-rank k forms a smooth manifold of dimensionality $(m+n-k)k$.
- Tangent space of the same dimensionality.
- Algorithm closely resembles a typical non-linear CG algorithm with Armijo line-search for unconstrained optimization.

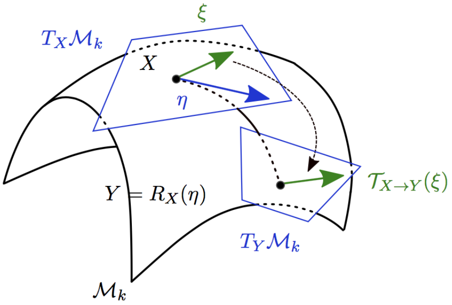
Skoltech

Visualization of non-linear CG on a Riemannian manifold



Skoltech

Vector transport on a Riemannian manifold



Initial formulation:

$$\underset{Z}{\text{minimize}} \quad \mathrm{rank}(Z) \quad \text{s.t.} \quad \frac{1}{2} \sum_{(i,j)\in\Omega} (X_{ij} - Z_{ij})^2 \leq \delta \qquad (1)$$

Rank minimization

Initial formulation:

$$\underset{Z}{\text{minimize}} \quad \mathrm{rank}(Z) \quad \text{s.t.} \quad \frac{1}{2} \sum_{(i,j) \in \Omega} (X_{ij} - Z_{ij})^2 \leq \delta \qquad (1)$$

Convex relaxation of (1):

$$\underset{Z}{\text{minimize}} \quad \|Z\|_* \quad \text{s.t.} \quad \frac{1}{2} \sum_{(i,j) \in \Omega} (X_{ij} - Z_{ij})^2 \leq \delta \qquad (2)$$

Rank minimization

Initial formulation:

$$\underset{Z}{\text{minimize}} \quad \mathrm{rank}(Z) \quad \text{s.t.} \quad \frac{1}{2}\sum_{(i,j)\in\Omega}(X_{ij}-Z_{ij})^2 \le \delta \qquad (1)$$

Convex relaxation of (1):

$$\underset{Z}{\text{minimize}} \quad \|Z\|_* \quad \text{s.t.} \quad \frac{1}{2}\sum_{(i,j)\in\Omega}(X_{ij}-Z_{ij})^2 \le \delta \qquad (2)$$

Equivalent reformulation of (2):

$$\underset{Z}{\text{minimize}} \quad \frac{1}{2}\sum_{(i,j)\in\Omega}(X_{ij}-Z_{ij})^2 + \lambda\|Z\|_* \qquad (3)$$

Skoltech

If the full X is known and $U\Lambda V^T$ is SVD for X, then the solution to (3) is given by

$$Z = U\Lambda_\lambda V^T, \text{ where } \Lambda_\lambda = \mathrm{diag}\Big((\sigma_1 - \lambda)_+, \ldots, (\sigma_{\min\{M,N\}} - \lambda)_+\Big).$$

SOFT-INPUT

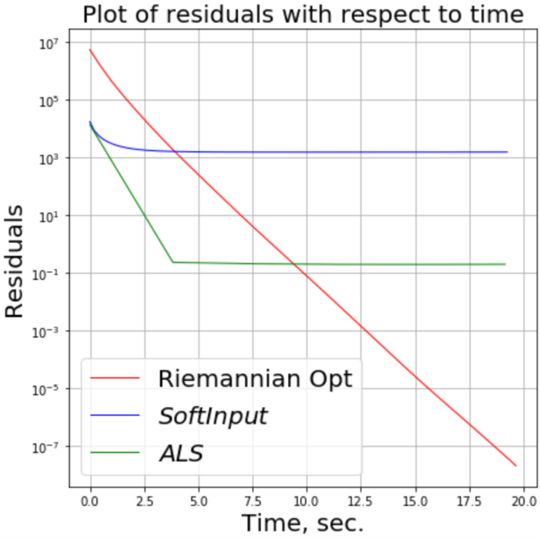
- Set Z_0 to be a zero-matrix.
- At each step k approximate unknown entries of X by Z_{k-1}, set Z_k to be a solution for a problem with all values given.

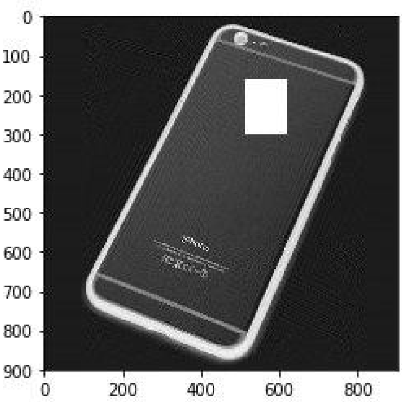
NOTE: Since approximated full matrix is of form $Z_k + (X - Z_{\Omega,k})$, MATVEC can be cheap.

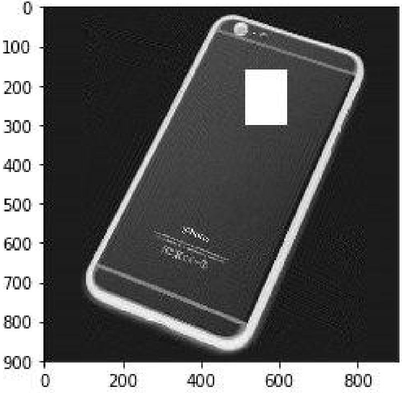
Comparative Analysis

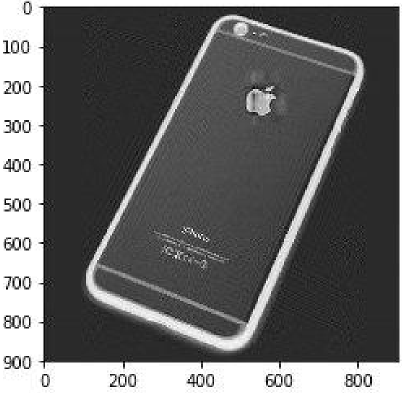
	ALS	**Riemannian**	**SOFT-INPUT**
Parallelizable	✓	✓	✗
Adaptive rank choice	✗	✗	✓
Flops per iteration	$O(\lvert\Omega\rvert K^2 + K^3)$	$O((M+N)K^2 + \lvert\Omega\rvert K))$	$O((M+N+\lvert\Omega\rvert)K^3)$

Comparative Analysis

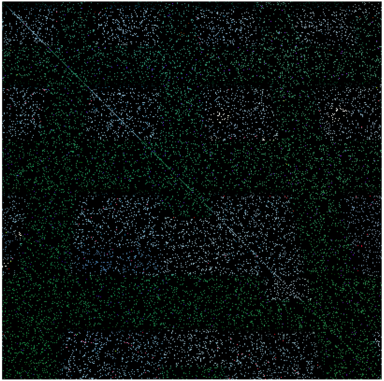



0
100
200
300
400
500
600
700
800
900
0
200
400
600
800

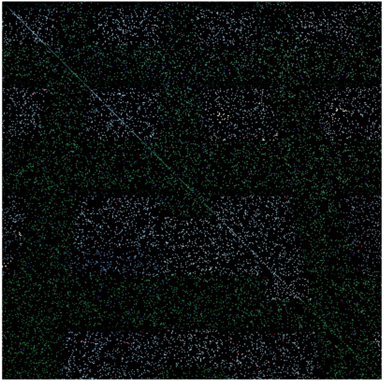

0
100
200
300
400
500
600
700
800
900
0
200
400
600
800

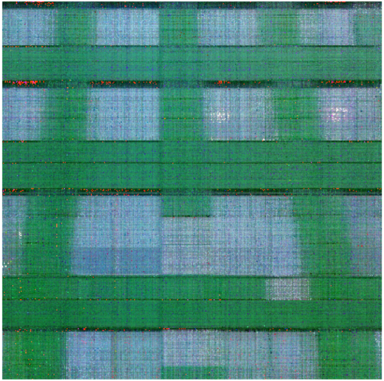

0
100
200
300
400
500
600
700
800
900
0
200
400
600
800

Demonstration

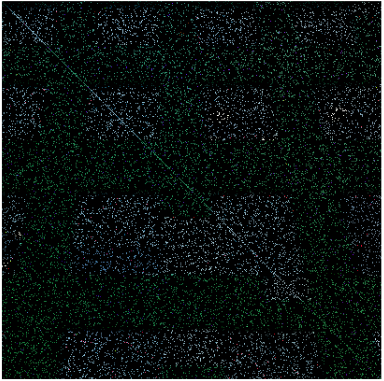


Demonstration

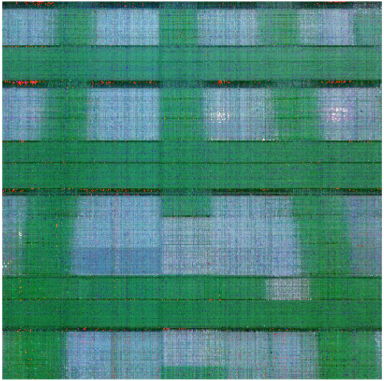




Demonstration



75 kB



451 kB

- Implementation of three competitive completion algorithms
- Comparative analysis of implemented algorithms
- Application to repairing damaged files