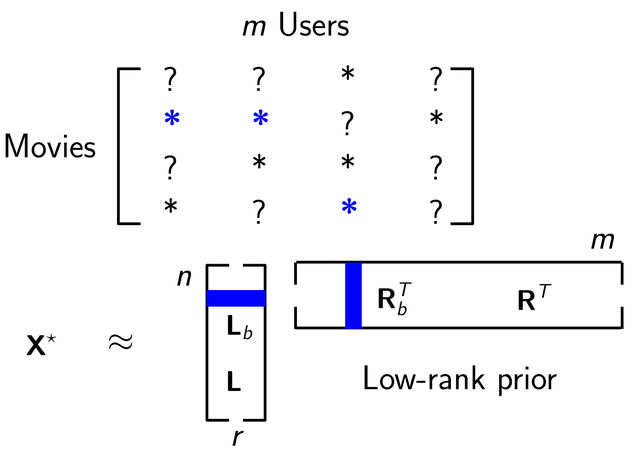
Scaled stochastic gradient descent for low-rank matrix completion

Bamdev Mishra[⋆] and Rodolphe Sepulchre[†]

[⋆]Amazon Development Centre India, Bangalore, Karnataka 560055, India (Bamdevm@amazon.com)

[†]University of Cambridge, Department of Engineering, Cambridge CB2 1PZ, UK (R.Sepulchre@eng.cam.ac.uk)

Motivation



$r \ll m, n$.

Formulation

$$\min_{\mathbf{X}\in\mathbb{R}^{n\times m}} \ \frac{1}{2}\|\mathcal{P}_\Omega(\mathbf{X}) - \mathcal{P}_\Omega(\mathbf{X}^\star)\|_F^2 \quad \text{subject to} \ \ \operatorname{rank}(\mathbf{X}) = r.$$

- The rank constrained is parameterized as $\mathbf{X} = \mathbf{L}\mathbf{R}^T$.
- Equivalently, the problem that is tackled by stochastic gradient descent (SGD) is

$$\min_{\mathbf{X}\in\mathbb{R}^{n\times m}} \ \frac{1}{2}\sum_{ij\in\Omega}(\mathbf{L}_i\mathbf{R}_j^T - \mathbf{X}^\star{}_{ij})^2.$$

- Ω is the set of known entries.

Contributions

- We propose a scaled variant of SGD that accelerates the standard SGD algorithm.
- We show that the proposed updates respect the scale invariance of the factorization model.
- The proposed updates are scalable.
  The computational complexity per epoch is $O(|\Omega|(r^3/b + b_{\mathbf{L}}r^2/b + b_{\mathbf{R}}r^2/b + r + \log b))$, where b is the batch size.
- Code: www.bamdevmishra.com/codes/scaledSGD.

Proposed scaled SGD algorithm

1. Pick b known entries with their indices.
2. Set up the completion subproblem by finding the indices corresponding to the submatrices $\mathbf{L}_b$ and $\mathbf{R}_b$, which need to be modified. Consequently, find the subset Ω_b of indices out of the total $b_{\mathbf{L}}b_{\mathbf{R}}$ indices.
3. Compute the residual $\mathbf{S}_b = \mathcal{P}_{\Omega_b}(\mathbf{L}_b\mathbf{R}_b^T - \mathbf{X}_b^\star)$.
4. Given a step-size t, update $\mathbf{L}_b$ and $\mathbf{R}_b$ as

$$\mathbf{L}_{b+} = \mathbf{L}_b - t\mathbf{S}_b\mathbf{R}_b\big(\tfrac{b\mu}{\max(m,n)}(\mathbf{R}^T\mathbf{R}) + (1-\mu)(\mathbf{R}_b^T\mathbf{R}_b)\big)^{-1}$$
$$\mathbf{R}_{b+} = \mathbf{R}_b - t\mathbf{S}_b^T\mathbf{L}_b\big(\tfrac{b\mu}{\max(m,n)}(\mathbf{L}^T\mathbf{L}) + (1-\mu)(\mathbf{L}_b^T\mathbf{L}_b)\big)^{-1}.$$

5. Update $\mathbf{L}^T\mathbf{L}$ and $\mathbf{R}^T\mathbf{R}$.
6. Repeat.

Choice of μ

- μ combines complementary local with global curvature information.
- $\mu = 0$ implies we use only local curvature information.
- $\mu = 1$ implies we use only global curvature information.
- In problem instances where a large number of entries are already known, i.e., $|\Omega|$ is large, the influence of $\mu < 1$ is minimal. However, for ill-conditioned data, making use of local information is more critical, and a smaller value of μ is more appropriate, e.g., $\mu = 0.5$.

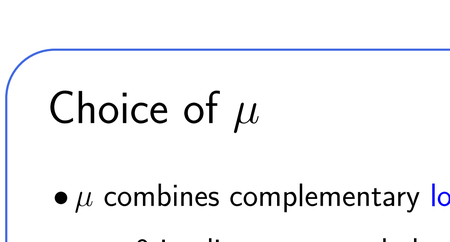
Choice of b

For $b = |\Omega|$, the proposed algorithm in behaves like a batch gradient descent algorithm. The choice of $b = 1$ is more appropriate for a fully online setting.

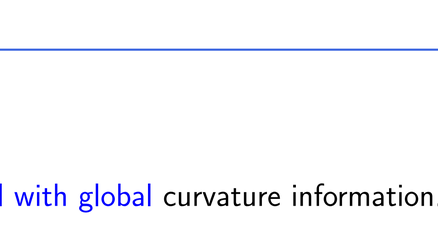
Scale invariance b

The proposed updates also resolve the issue of scale invariance arising from non-uniqueness of matrix factorization. The issue refers to the behavior of algorithms which should behave *equivalently* when initialized, say, either with $(\mathbf{L}_0, \mathbf{R}_0)$ or with $(\mathbf{L}_0\mathbf{M}^{-1}, \mathbf{R}_0\mathbf{M}^T)$ for all non-singular matrices $\mathbf{M}$. This is achieved by our updates.

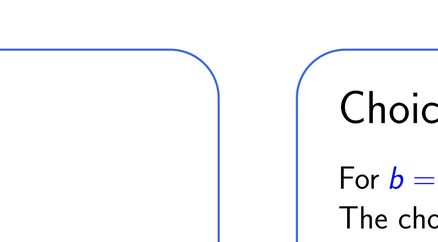
Numerical comparisons



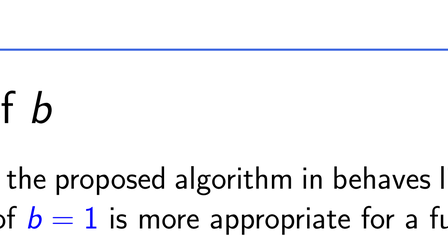
(a) Scale invariance.



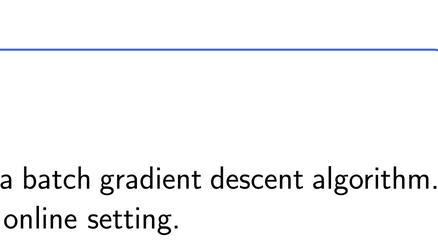
(b) Effect of μ on well-conditioned data.



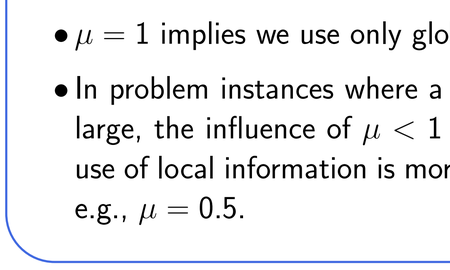
(c) Effect of μ on data with CN = 50.



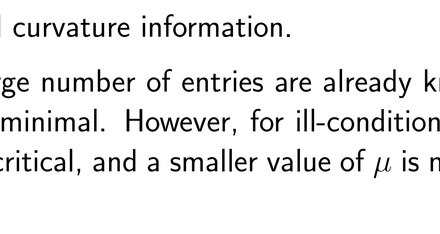
(d) Effect of b on well-conditioned data.



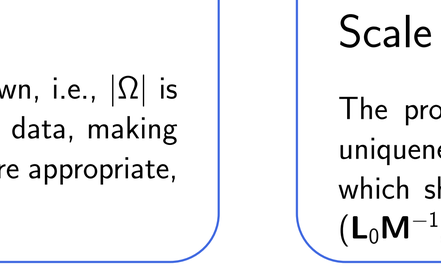
(e) Low-sampling with OS = 2.1.



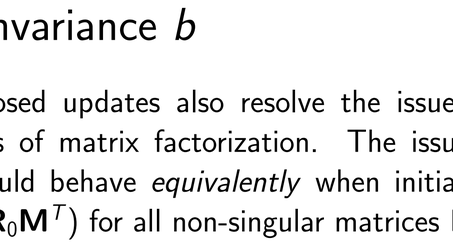
(f) Noisy instance.



(g) Ill-conditioned data with CN = 100.



(h) Rectangular instance.



(i) Jester dataset.

Algorithm	$n = 2000$		$n = 5000$		$n = 24983$	
	$r = 5$	$r = 7$	$r = 5$	$r = 7$	$r = 5$	$r = 7$
Scaled-SGD	0.158	0.159	0.160	0.158	0.159	0.157
SGD	0.158	0.159	0.160	0.158	0.159	0.157
Grouse	0.165	0.165	0.179	0.177	0.168	0.166
Loreta	0.159	0.160	0.160	0.158	0.159	0.158
ALS	0.158	0.159	0.160	0.158	0.159	0.157
CCD++	0.158	0.159	0.160	0.157	0.159	0.157

(j) Final mean NMAEs obtained on the test set of the Jester dataset.

This work was initiated while the first author was with the Department of Electrical Engineering and Computer Science, University of Liège, 4000 Liège, Belgium and was visiting the Department of Engineering (Control Group), University of Cambridge, Cambridge CB2 1PZ, UK.