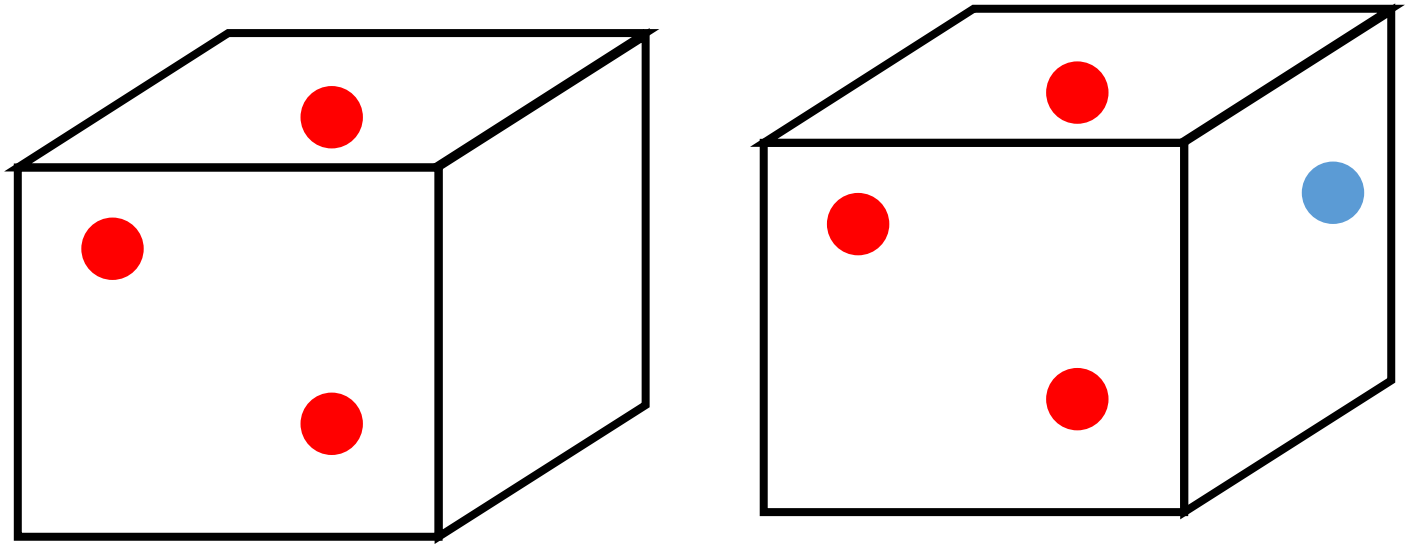
Online CP

Zimingliu

Online learning



t $t+1$

Each time, take in one new sample, and update the model according to the given sample.

Algorithm

Loss function
- Constrained or Non-constrained √
- Regularization √
- Recursive Least Squares (≈Momentum SGD)
- Sparse term (Robust high-order PCA)

Optimization technique
- SGD √
- SGD-newton (Quasi-newton)
- SGD + momentum (Nesterov, Adam,……)
- SVRG ???
- Backtracking

Online CP (Algorithm)

Algorithm 1 Online CP decomposition via stochastic gradient descent

N-way tensor $\mathcal{Y}$ has shape $I_1 \times I_2 \times \cdots \times I_N$

Note: We aim at $\min_{\{A^{(i)}\}} ||\mathcal{Y} - [[A^{(1)}, \cdots, A^{(N)}]]||_F^2 + \sum_{i=1}^{N} \frac{\mu}{2} ||A^{(i)}||_F^2$

Initialize factor matrices $A^{(i)}$

while $k = 1 : \text{iter}$ **do**

  while $i = 1 : N$ **do**

    Loss function becomes $\min_{A^{(i)}} ||Y^{(i)} - A^{(i)}(\odot_{j \neq i} A^{(j)})^T||_F^2 + \frac{\mu}{2} ||A^{(i)}||_F^2$

    Calculate Khatri-Rao Product $K^{(i)} = (\odot_{j \neq i} A^{(j)})^T$

online: Random choosing (m, n) element from $Y^{(i)}$ $(m \in 1{:}I_i, n \in 1{:}\Pi_{j \neq i} I_j)$

    Least squares gradient g^{ls} =zeros$(I_i, \Pi_{j \neq i} I_j)$

    $g^{ls}(m, :) = (A^{(i)}(m, :)K^{(i)}(:, n) - Y^{(i)}(m, n))K^{(i)T}(n, :)$

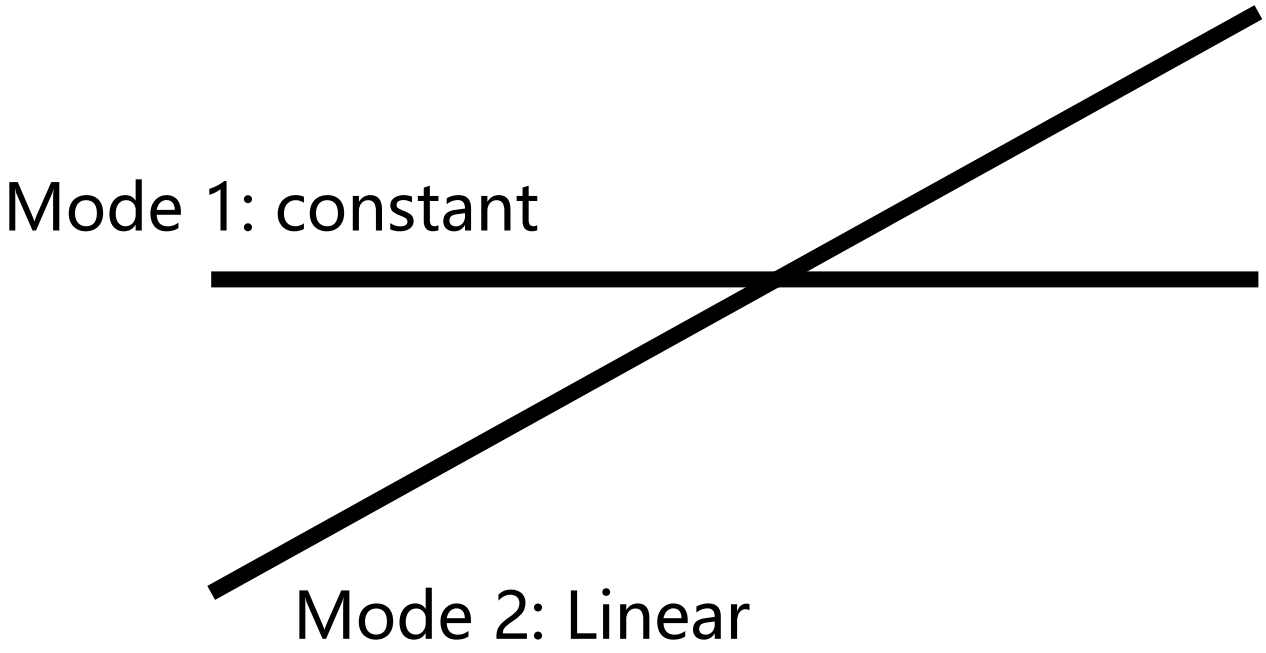
gradient: Regularization gradient $g^{reg} = \mu A^{(i)}$

    Total gradient $g = g^{ls} + g^{reg}$

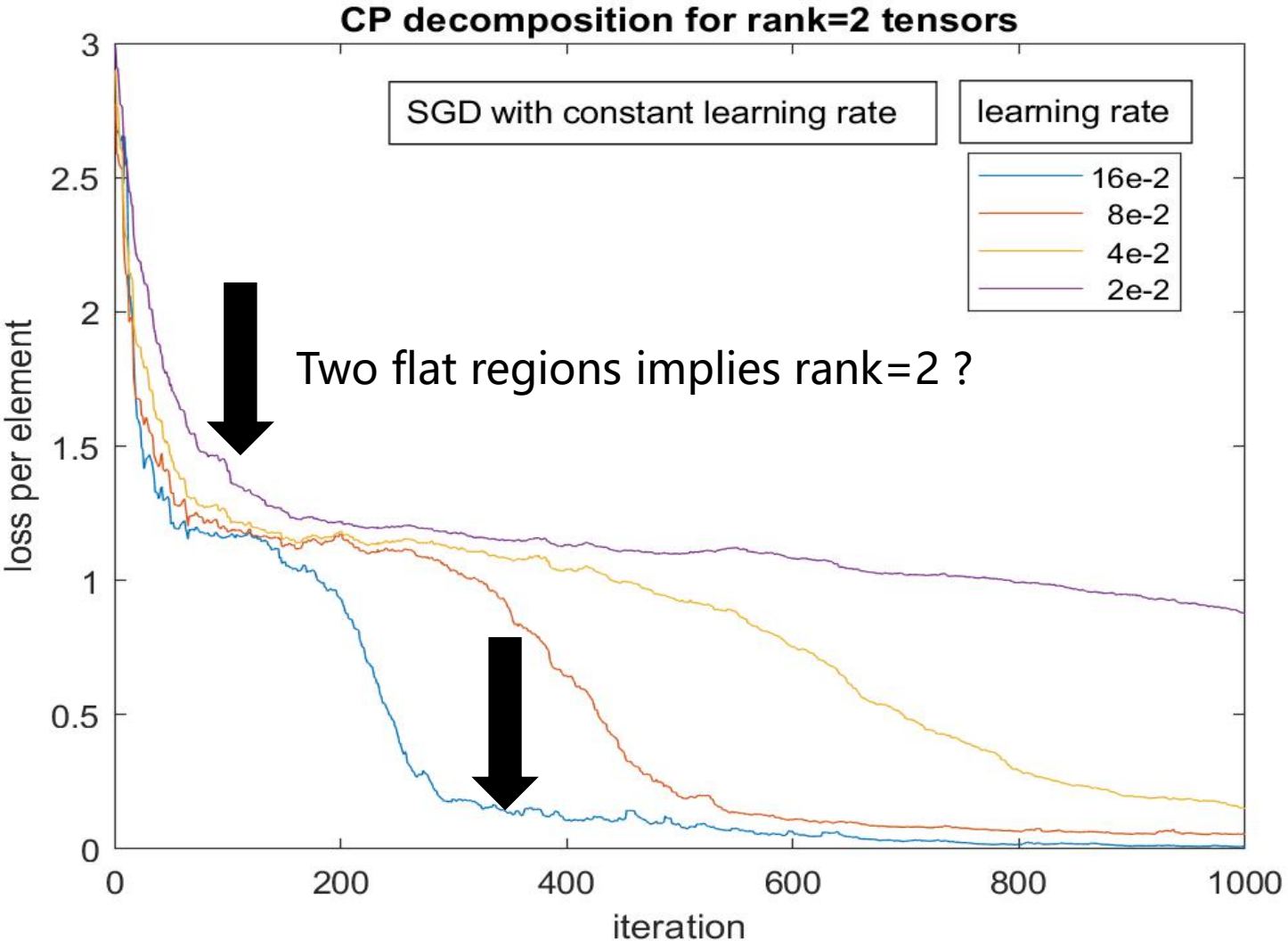
    Update $A^{(i)}$ with gradient: $A^{(i)} \leftarrow \text{anymethod}(A^{(i)}, g)$

Mock data

N=3, 5*5*5 tensor
Rank=2, symmetric

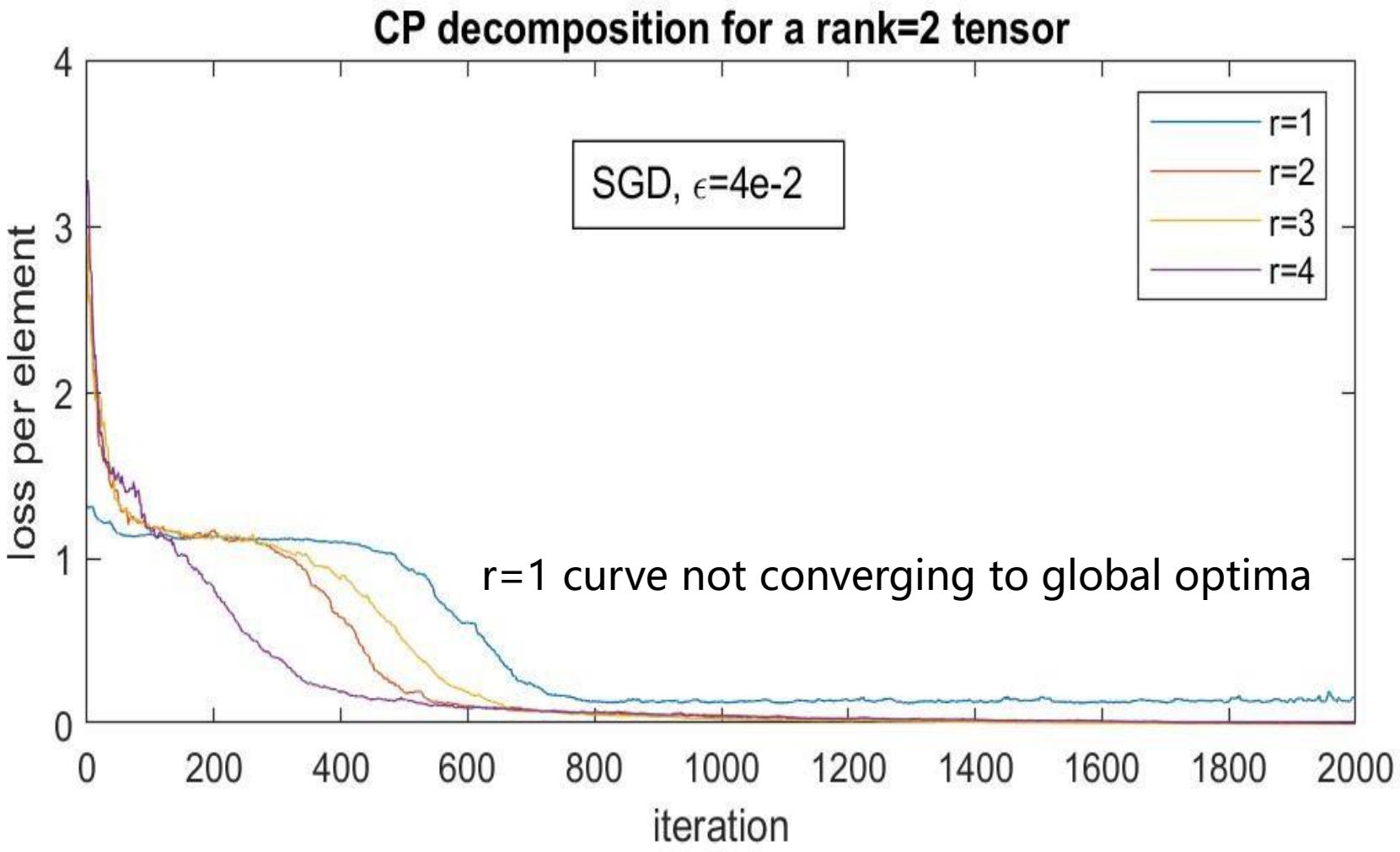


SGD, learning rate



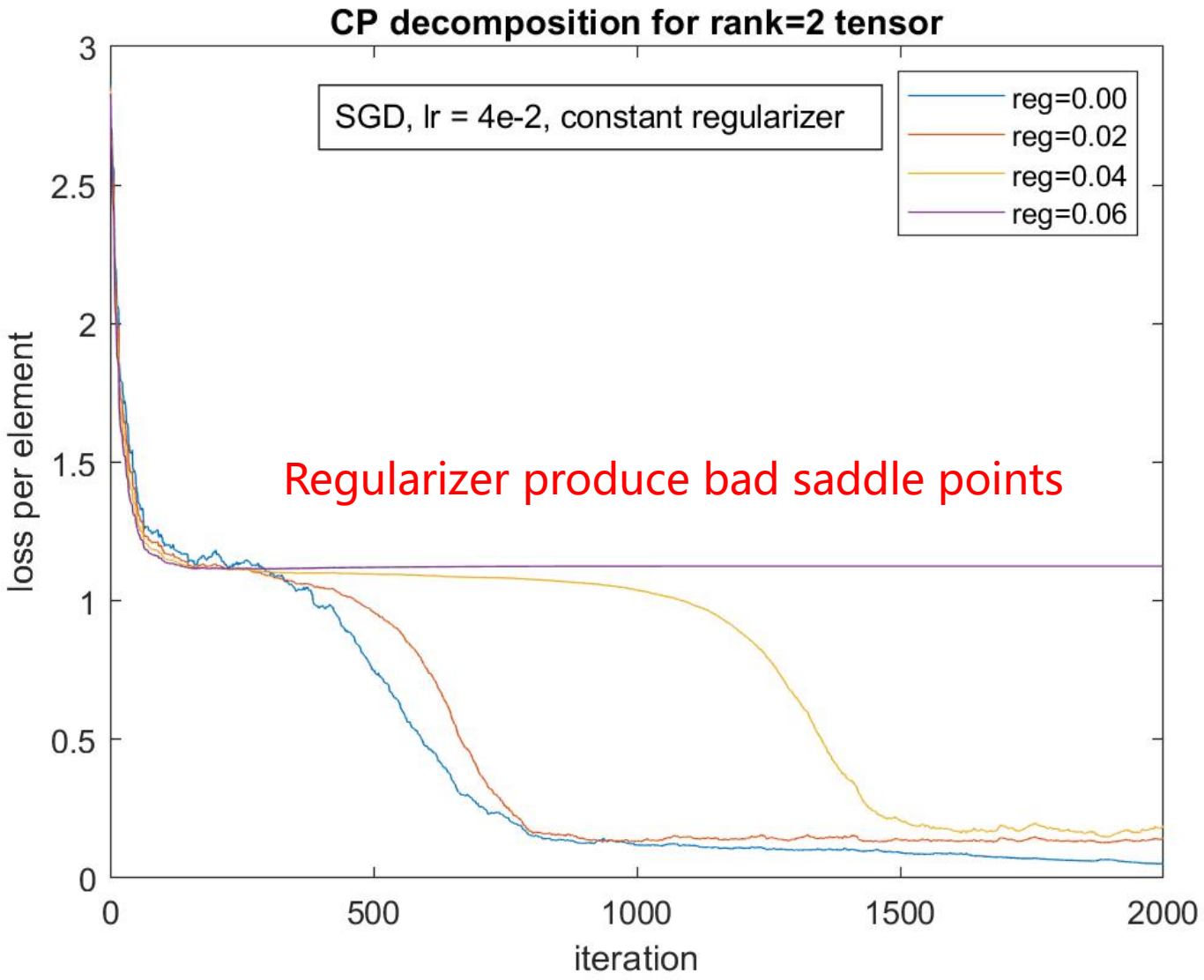
At early steps, large learning rate is preferable.

Estimated rank

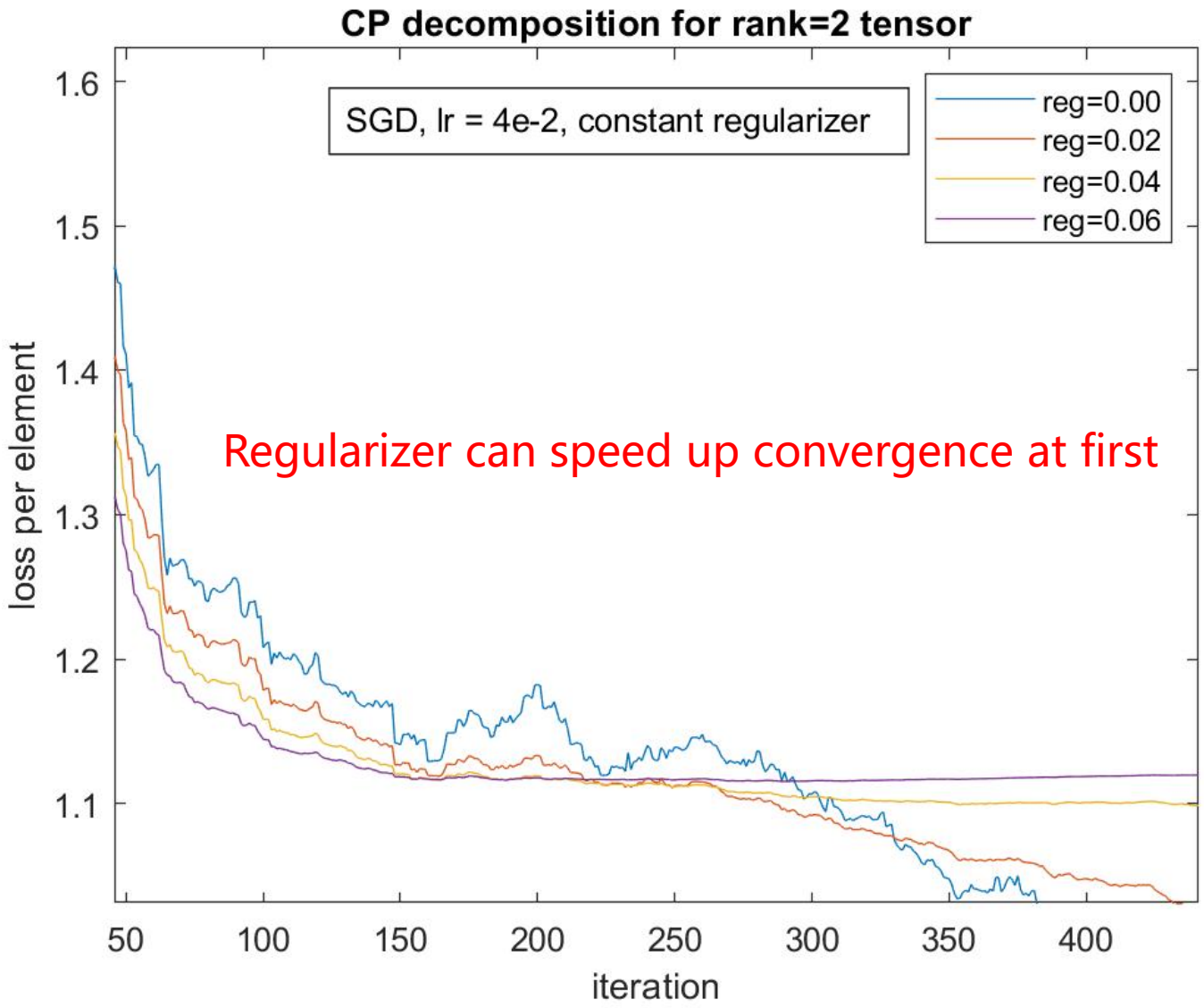


Overestimating rank is not a serious problem.

Regularization

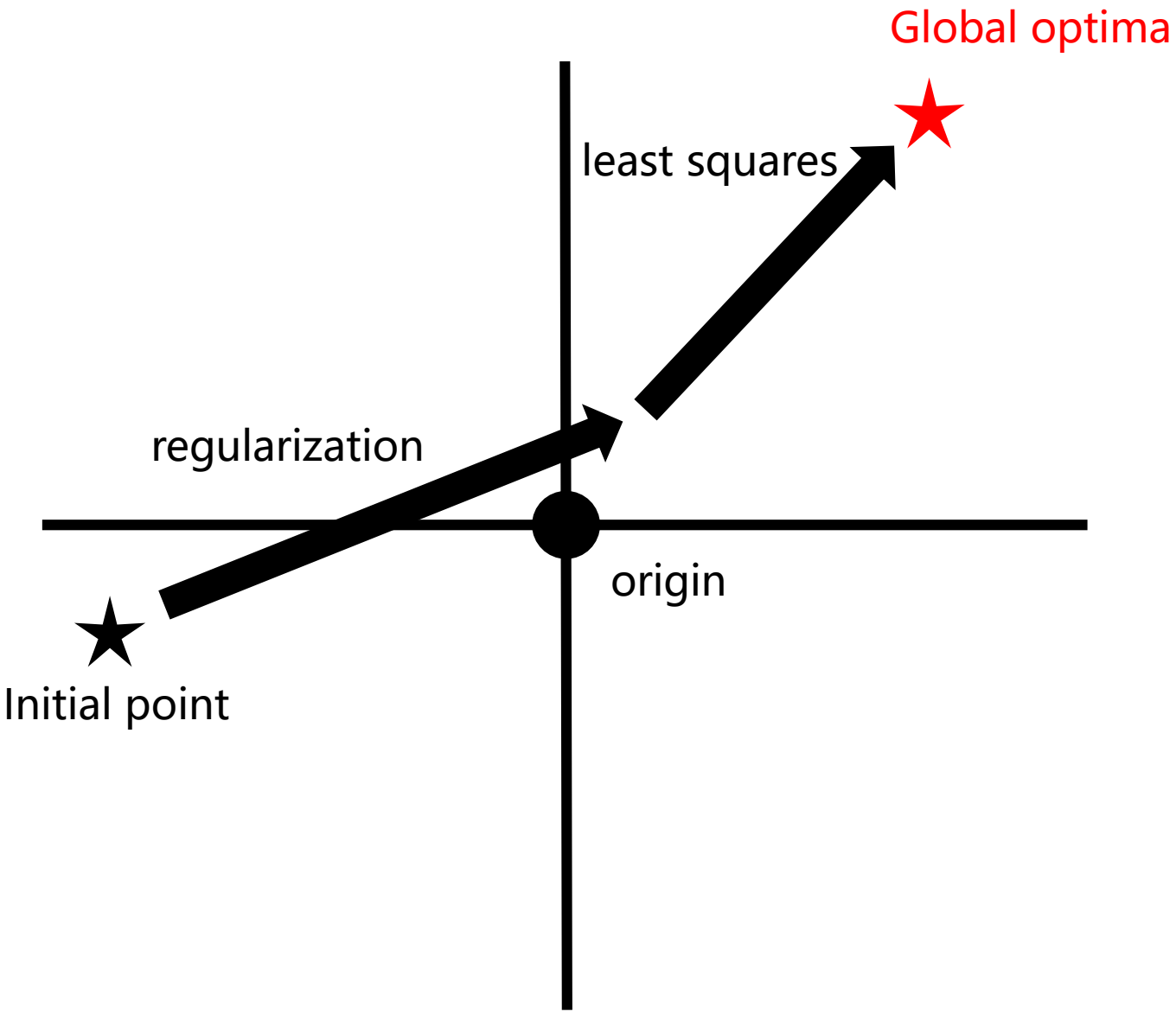


Regularization

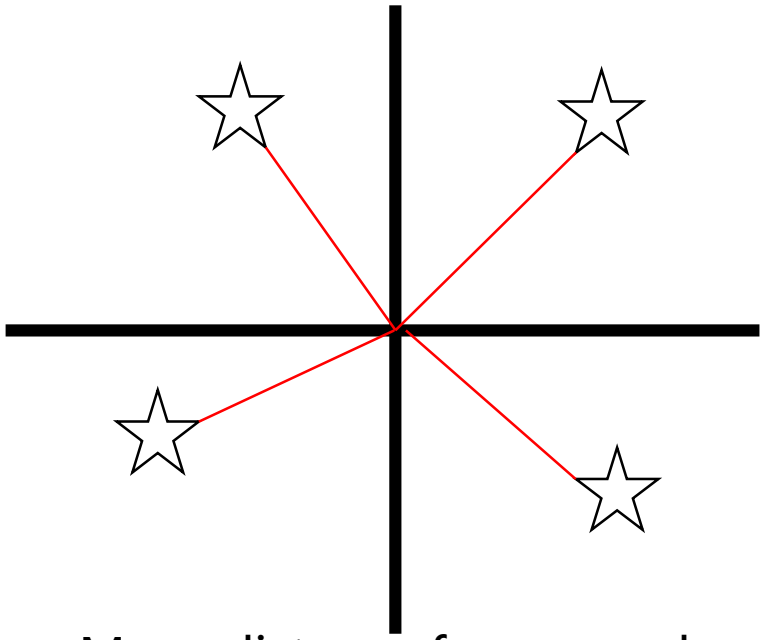


Q: We can try *adaptive* regularization?

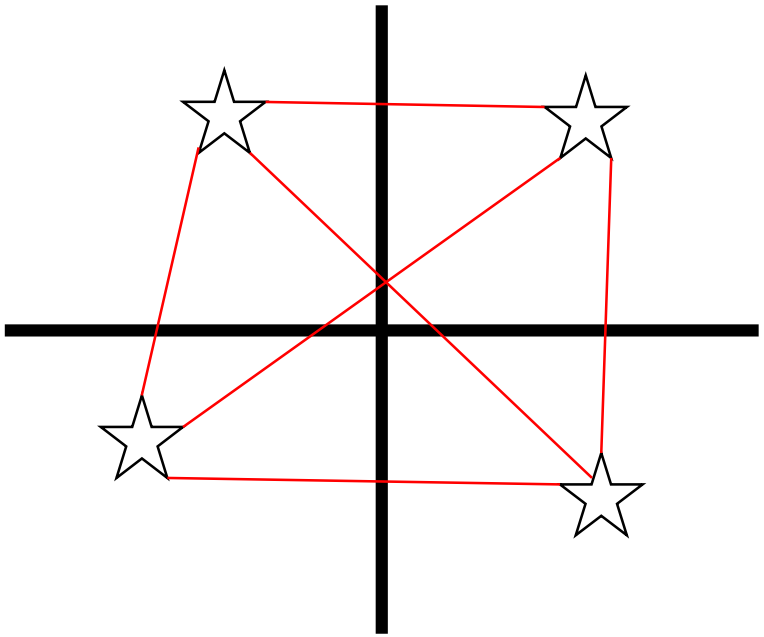
Possible Explanation



Possible Explanation



Mean distance from sample to origin: r



Mean distance from sample to sample : $\sqrt{2}r$

Backtracking

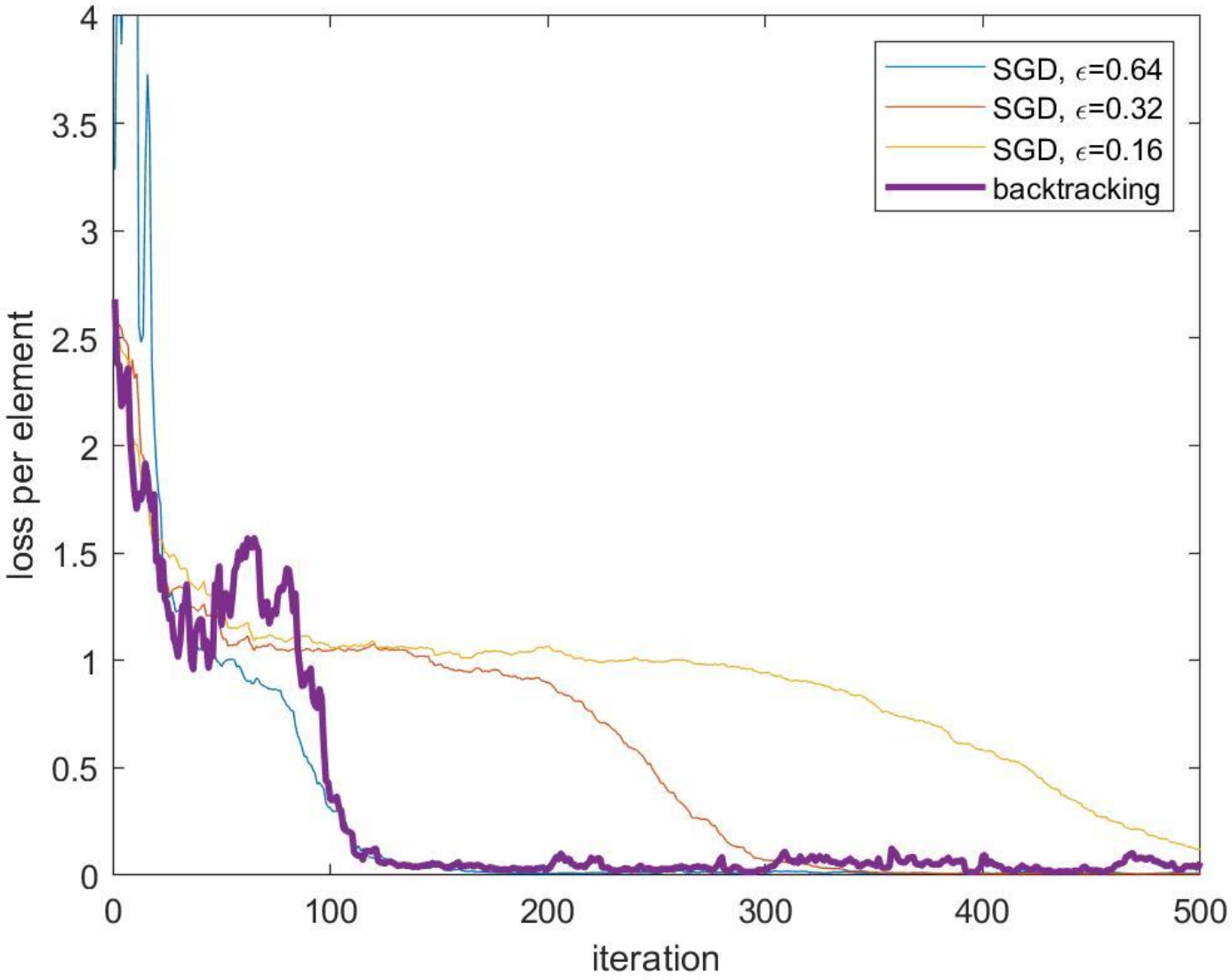
A way to adaptively choose the step size

- First fix a parameter $0 < \beta < 1$
- Then at each iteration, start with $t = 1$, and while

$$f(x - t\nabla f(x)) > f(x) - \frac{t}{2}\|\nabla f(x)\|^2,$$

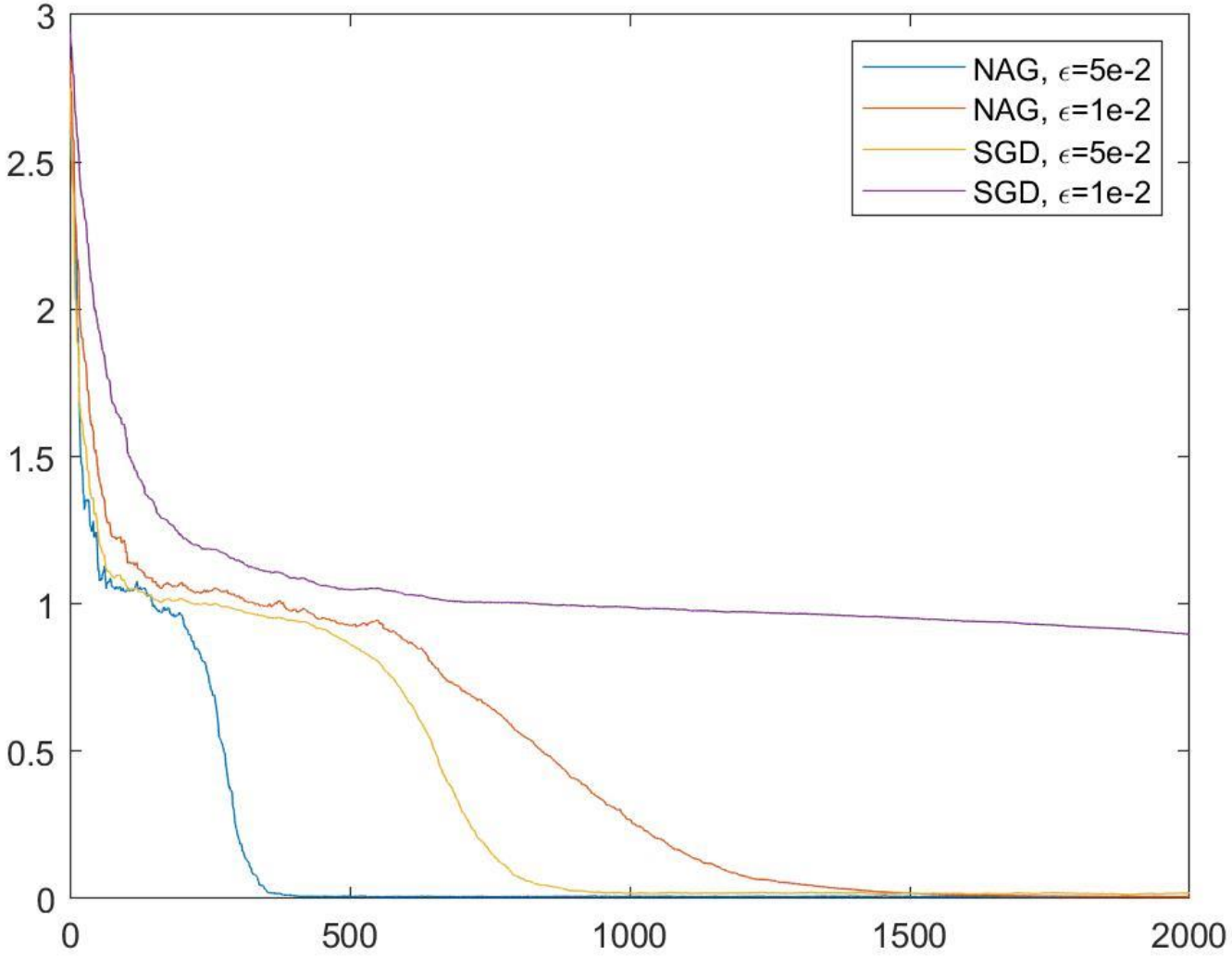
update $t = \beta t$

Backtracking



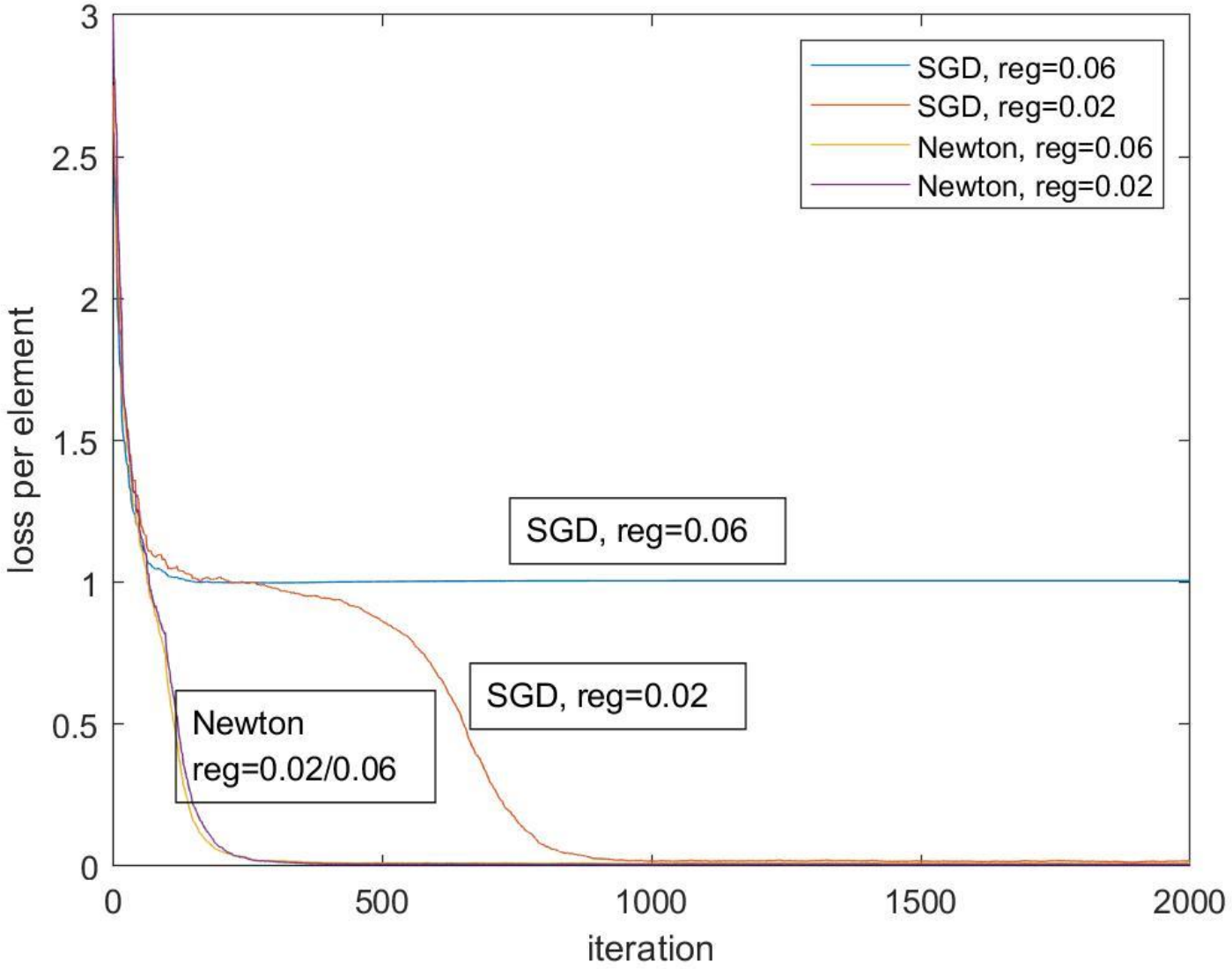
Backtracking adaptively changes step sizes, no need to fine tune !

Momentum (Nesterov)



Momentum helps escaping from saddle points faster!

Second order method



Newton’s method converges faster than SGD method, and not sensitive to amount of regularization.

Online CP problem (Plan)

Online CP

- Real-life dataset ?
- Techniques to speed up convergence
- Theoretical foundations

Online CP paper review

Online CP literature review

Problem Paper

[2] Dynamical Subspace; [3] Sparse tensor; [4] Background;
[5] Constraint (non-negativity); [8] Concept drift, like [2];
[9] Orthogonal Tensor; [15] General CP

Method Paper

[1] high order; [2] RLS; [6] Parallelizable sparse;
[7] Randomized block sampling; [10] SGD, 2SGD, SALS;
[11] Randomized sampling; [12] Sketching (hashing);
[14] SamBaTen, similar to [7]; [15] Stochastic sampling

Theory Paper (# of samples; Global/Local cvg)

[9] Saddle Point Theory;
[13] Moment tensor, convergence rate ……

1. Accelerating Online CP Decompositions for Higher Order Tensors

Problem:
CP streaming
ALS poor scalability
Only 3-order method exist

Proposed:
High-order method
Avoid duplication computation of Khatri-Rao products

Theory:
None
Method:
Compared with **SDT and RLST** and **GridTF**
Experiment:
Image (COIL, FACE), human act(DSA,HAD,FOG),chem(GAS),traffic(ROAD)

2. Fast online low-rank tensor subspace tracking by CP decomposition using recursive least squares from incomplete observations (OLSTEC)

Problem:
CP streaming
Incomplete + noise
Subspace change over time
Processing speed>data acquiring speed, we care convergence rate (RLS)

Proposed:
Convex relaxation -- poor scalability; direct non-convex optimization
Mini-batch -- Batch-based method fail; Online

Theory:
RLS / momentum method / second-order : no theory
Method:
(Matrix based)
PAST(projection), GROUSE(Grassman), GRASTA(Robust),PETRELS(parallel)
(Tensor based)
19 CP, 20 Tucker (ALTO), 21 CP+missing+SGD (slow), 22 Riemannian preconditioning. All first-order method
Experiment:
Synthetic, traffic, environment, surveillance videos

3. Online CP Decomposition for Sparse Tensors

Problem:
Sparse
CP

Proposed:
[9] 3-order dense (SDT and RLST); [10] high-dim (1) dense
[17] sub-sampling, both sparse and dense [18][19] add one element
[20] increase at all modes [This paper] add one slice (time)

Theory:
None
Method:
Very much like [10], but use sparse operations
Experiment:
Facebook-Links; Facebook-Wall; MovieLens; LastFM; NIPS; Youtube; Enron; NELL-2; NELL-1

4. Online Stochastic Tensor Decomposition for Background Subtraction in Multispectral Video Sequences

Problem:
Dense 3-d tensor (2d image*multi-spectra)
Background subtraction

Proposed:
Using multi-spectral image
ALS-like method (low-rank+sparse)
Convex relaxation

Theory:
None
Method:
ALS-like
Experiment:
Synthetic data+ multi-spectral video

5. Streaming Tensor Factorization for Infinite Data Sources (CP stream)

Problem:
Constraint, memory-efficient (previous work: linear in time)

Proposed:
ALS-like
Not update time domain as a whole each time (complexity: T->1)

Theory:
None
Method:
ALS-like
Experiment:
AirportHall, ChicagoCrime, Reddit2008, CyberLANL

6. ParCube: Sparse Parallelizable Tensor Decompositions

Problem:
A tensor too large (cannot even fit in the memory)
CP

Proposed:
Sample sub-tensors, do CP
Merging modes for sub-tensors

Theory:
None
Method:
Subtensor, similarity matching
Experiment:
ENRON, LBNL Network Traffic, Facebook Wall Posts, NELL

7. A randomized block sampling approach to canonical polyadic decomposition of large-scale tensors

Problem:
A tensor too large (cannot even fit in the memory)
CP

Proposed:
Sample sub-tensors
Update Corresponding CP factors

Theory:
None
Method:
Subtensor, Gradient+Hessian (Like Trust region method)
Experiment:
Synthetic data

8. Identifying and Alleviating Concept Drift in Streaming Tensor Decomposition

Problem:
Streaming tensor have changeable concepts

Proposed:
SeekandDestroy

Method:
Getrank + Modes matching

9. Escaping From Saddle Points – Online Stochastic Gradient for Tensor Decomposition

Problem:
SGD little theory guarantee

Proposed:
Add noise to the computation of gradient

Theory:
A lot. Global convergence of SGD on non-convex problems
Orthogonal tensor decomposition

10. Expected Tensor Decomposition with Stochastic Gradient Descent

Problem:
expected CP decomposition

Proposed:
SGD, 2SGD, SALS (Most related to ours)

Theory:
Convergence guarantee, but no convergence rate
Good:
Efficient memory use, work in an online setting, robustness of parameter tuning, simplicity

11. MACH: Fast Randomized Tensor Decompositions

Problem:
CP decomposition

Proposed:
Randomly draw elements from tensor (amplified by a magnitude)
others setting zero

Theory:
A lot : convergence rate
Good:
When measurement is expensive

12. Fast and Guaranteed Tensor Decomposition via Sketching

Problem:
CP decomposition

Proposed:
Tensor Power method
Hashing/sketching

Theory:
A lot : convergence rate
Experiment:
Synthetic + Topic Modelling

13. Online and Differentially-Private Tensor Decomposition

Problem:
CP decomposition

Proposed:
Tensor Power method
Moment Tensor, orthogonal tensor (Deflation is available)

Theory:
A lot : convergence rate

14. SamBaTen: Sampling-based Batch Incremental Tensor Decomposition

Problem:
CP decomposition

Proposed:
Subtensor
Matching condition

Theory:
No theory
Experiment:
A lot

15. Stochastic Gradients for Large-Scale Tensor Decomposition

Problem:
CP decomposition
GCP

Proposed:
Sampling strategy
Stochastic For dense tensor
GCP

Theory:
No theory
Experiment:
A lot

Back Up

Related works (Adaptive Reg)

Buchanan, Aeron M., and Andrew W. Fitzgibbon.
"Damped newton algorithms for matrix factorization with missing data."
2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). Vol. 2. IEEE, 2005.

Agarwal, Naman, et al.
"Efficient Full-Matrix Adaptive Regularization."
International Conference on Machine Learning. 2019.

Estellers, Virginia, Stefano Soatto, and Xavier Bresson.
"Adaptive regularization with the structure tensor."
IEEE Transactions on Image Processing 24.6 (2015): 1777-1790.

.....