Fast and Memory Optimal Low-Rank Matrix Approximation

Team #17

<u>Team members:</u>

Semyon Abramov
Denis Koposov
Daniil Lopatkin
Albert Nagapetyan
Viktor Prutyanov

<u>Mentor:</u>

Daniil Merkulov

Background

Low-rank approximation of matrix $M \in [0,1]^{m \times n}$ is a matrix **Z** such that $\min_{\text{rank}(Z)=k} \|M - Z\|_F$

Possible applications:

- Image compression
- Noise reduction
- Latent semantic indexing

Algorithm: Streaming Low-Rank Approximation (SLA)

Idea:

1. Independently sample entries from $\mathbf{M}$ to $\mathbf{A_1}$, $\mathbf{A_2}$
2. $\mathbf{Q}$ is $\mathbf{k}$-rank PCA for first $\mathbf{l}$ columns of $\mathbf{A_1}$
3. Trim some rows and columns of $\mathbf{A_2}$
4. Create sketch of the matrix based on basis $\mathbf{Q}$, remove $\mathbf{A_1}$, $\mathbf{A_2}$, $\mathbf{Q}$ from memory
5. Iteratively update sketch for every new column of $\mathbf{M}$

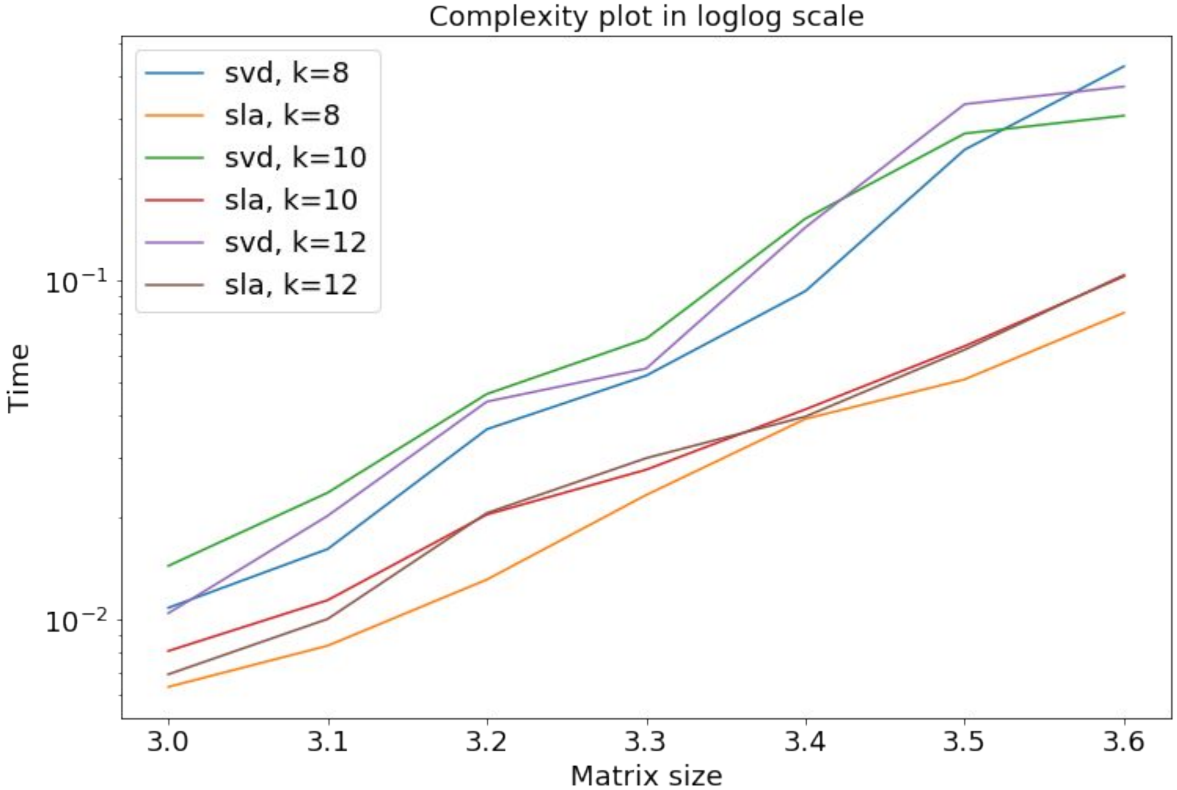
Key features:

- Streaming: matrix is observed in a sequential order in a single pass
- Only one observed column of size $\mathbf{m}$ is needed at each iteration
- Memory complexity is $\mathbf{O(k(n+m))}$

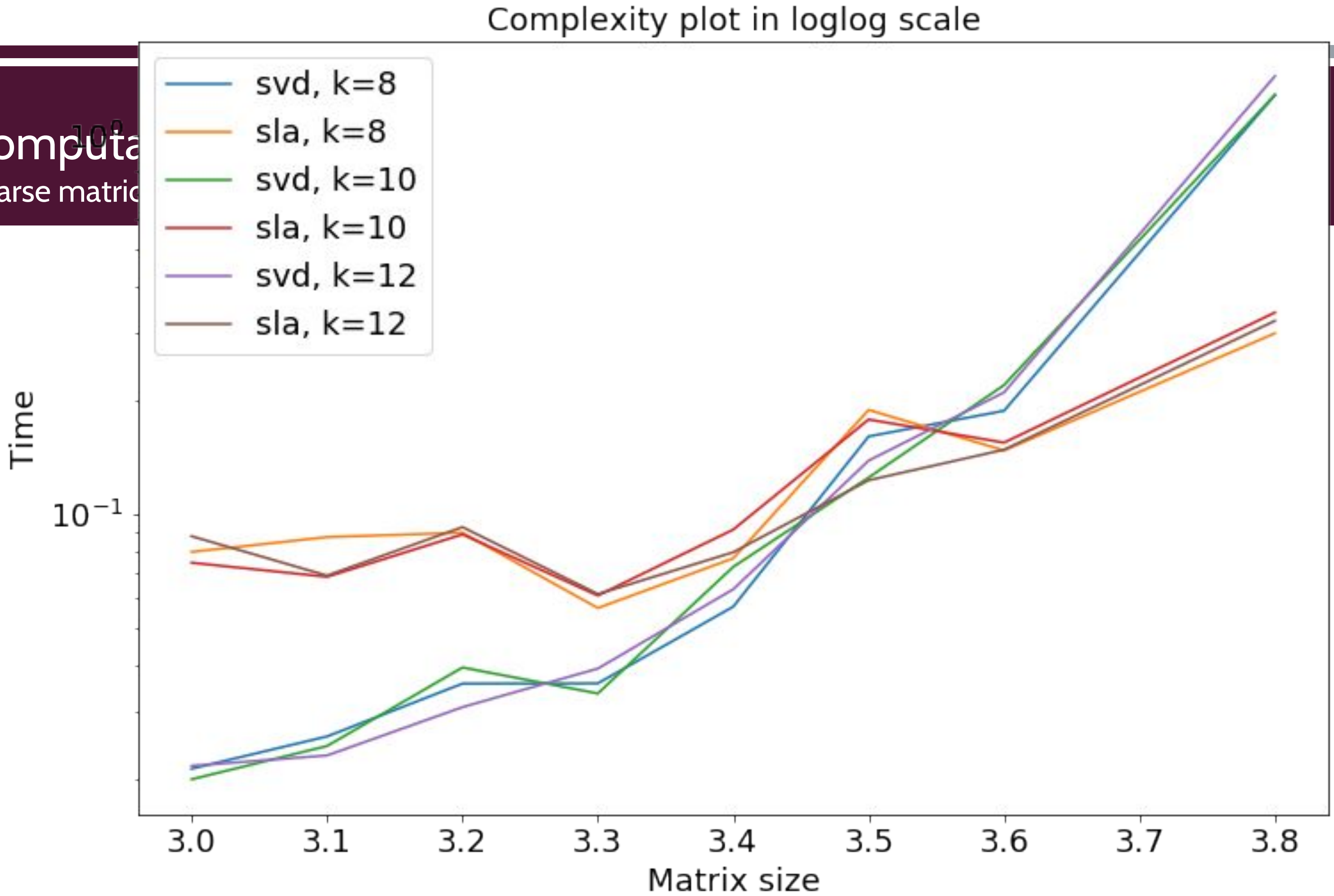
Se-Young Yun, Marc Lelarge, Alexandre Proutiere. [Fast and Memory Optimal Low-Rank Matrix Approximation]

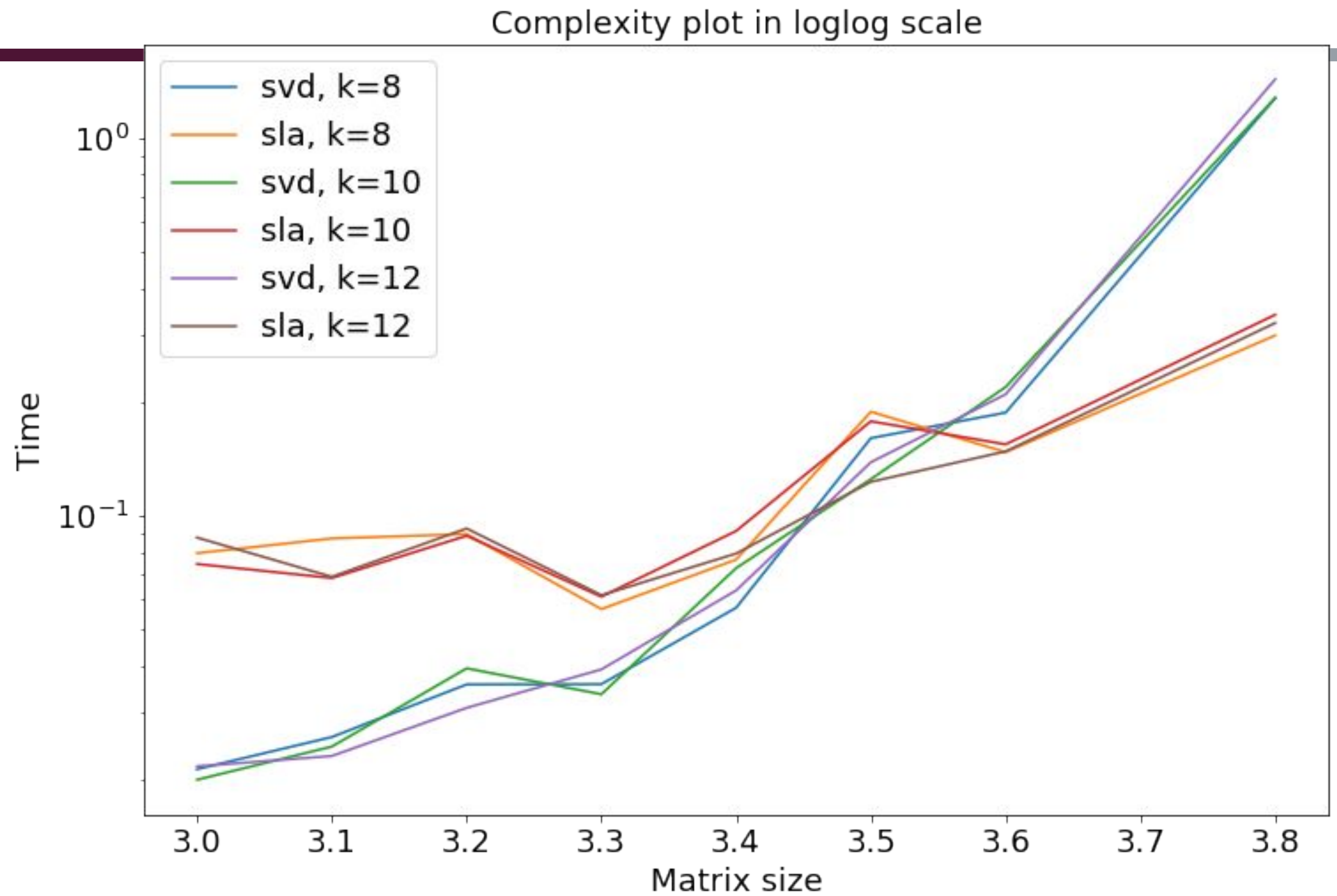
Computational complexity

Dense matrices

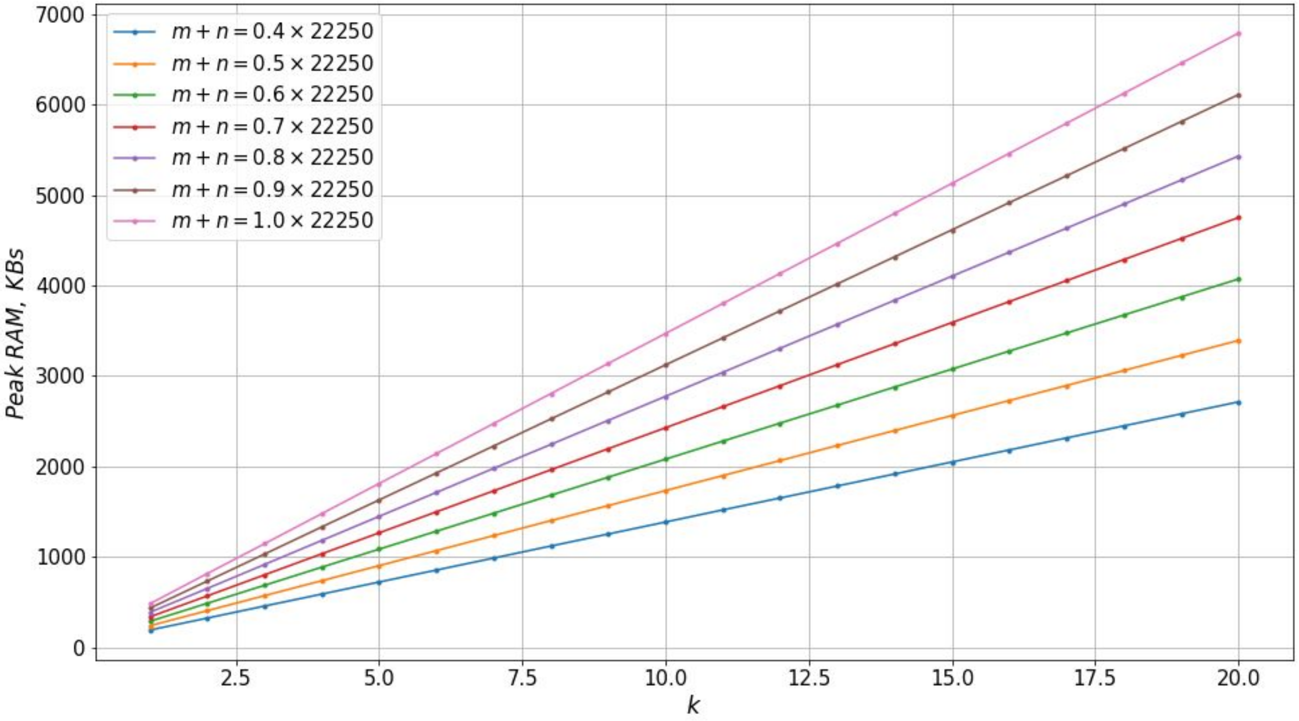


Computa…
Sparse matric…




Complexity plot in loglog scale
svd, k=8
sla, k=8
svd, k=10
sla, k=10
svd, k=12
sla, k=12
Time
10^0
10^{-1}
3.0
3.1
3.2
3.3
3.4
3.5
3.6
3.7
3.8
Matrix size

Memory consumption



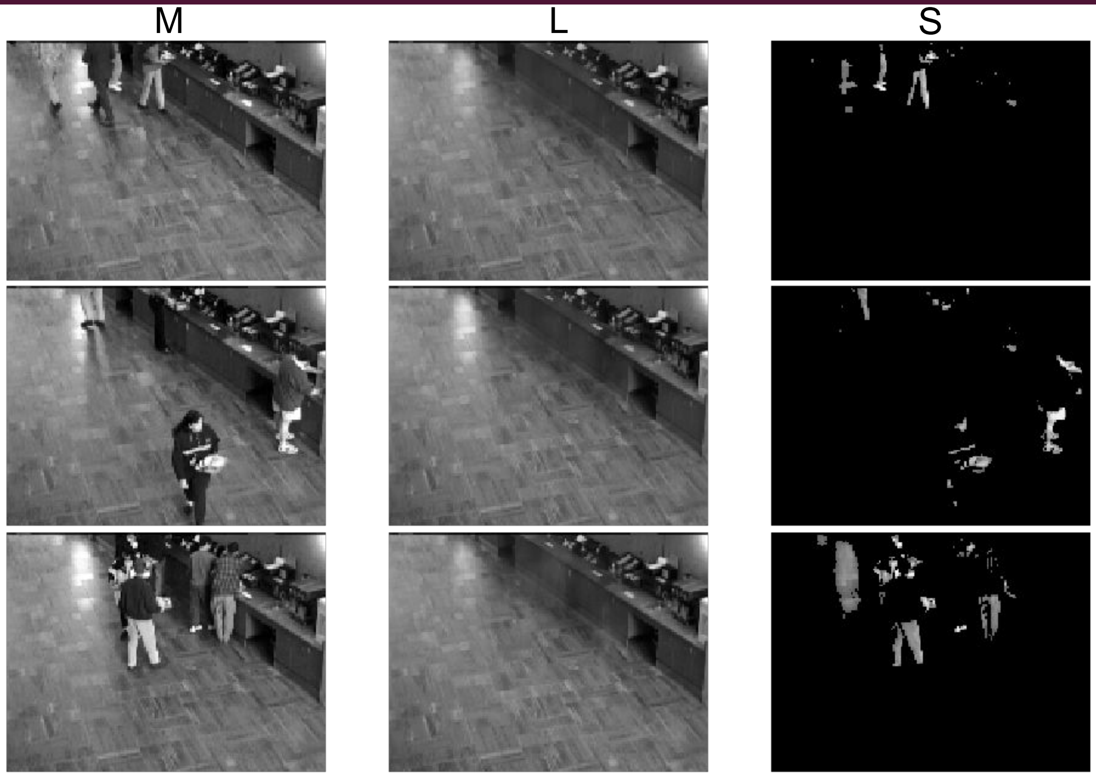
- All memory allocations for matrices were tracked
- Memory complexity is **`O(k(n+m))`** — as in theory

Applications

Video foreground/background separation

- `M = L + S`
- **L** — low-rank matrix
- **L** — static background (scene)
- **S** — sparse matrix
- **S** — foreground (moving objects)

- Video: [link]
- **~3000** frames
- **1-2** seconds to process



Dataset: K. Toyama. Wallflower: Principles and Practice of Background Maintenance

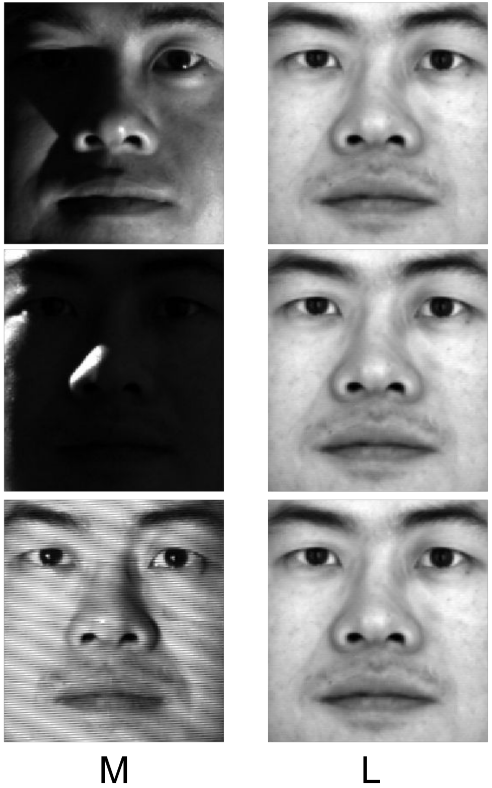
Applications

Shadow removing

- `M = L + S`, the same as in the previous slide
- 58 images of the same person
- Different illumination conditions
- ~ **`100ms`** per person

References:

- E. J. Candes. Robust PCA
  (~ 5 min for video, 1.5 min for faces)
- A. Yurtsever. Sketchy Decisions: Convex Low-Rank Matrix Optimization with Optimal Storage
  (~ 30 min for video)
- [Yale Face Database]



Conclusion

Streaming Low-Rank Approximation (SLA) method:

- Optimal memory consumption
- Real-time data processing
- Good performance in comparison with SVD, Robust PCA method, Sketchy Conditional Gradient Method (SketchyCGM)

Contribution

- Semyon Abramov
  - SLA implementation
  - Application: faces shadow removing
- Denis Koposov
  - SLA implementation
  - Application: faces shadow removing
- Daniil Lopatkin
  - SLA implementation
  - Application: video foreground/background separation
- Albert Nagapetyan
  - Performance analysis
- Viktor Prutyanov
  - Application: video foreground/background separation
  - Memory consumption analysis