Supplementary Material:Large-Scale Matrix Factorization with Missing Data under Additional Constraints

Kaushik Mitra
Department of Electrical and Computer Engineering and UMIACS
University of Maryland, College Park, MD 20742
`kmitra@umiacs.umd.edu`

Sameer Sheorey
Toyota Technological Institute, Chicago
`ssameer@ttic.edu`

Rama Chellappa
Department of Electrical and Computer Engineering and UMIACS
University of Maryland, College Park, MD 20742
`rama@umaics.umd.edu`

Abstract

We present results from our experiments that we could not include in the main paper [2].

1 Experimental Evaluation

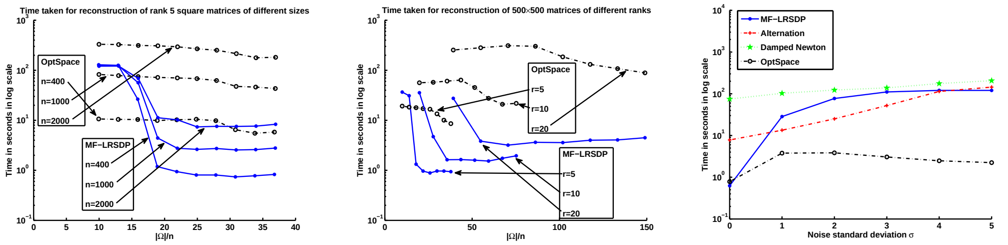
We compare the timing performance of MF-LRSDP, OptSpace, alternation and damped Newton for various synthetic and real experiments.

1.1 Evaluation with Synthetic Data

Exact Factorization: vary size. Figure shows the time (in seconds, log-scale) that MF-LRSDP and OptSpace takes for factorization of rank 5 matrices of different sizes n. Clearly, MF-LRSDP is more efficient.

Exact Factorization: vary rank. Figure shows the time (in seconds, log-scale) that MF-LRSDP and OptSpace takes for factorization of 500×500 matrices of different ranks r. Again, MF-LRSDP takes lesser time for reconstruction.

Noisy Factorization: vary noise standard deviation. We vary the standard deviation σ of the additive noise for rank 5, 200×200 matrices and study the timing performance by MF-LRSDP, OptSpace, alternation and damped Newton. MF-LRSDP takes more time than OptSpace because the noise terms are treated as variables in MF-LRSDP. This might seem like a disadvantage, but it also makes MF-LRSDP more flexible as we can, potentially, minimize the L_1 norm of noise instead of the L_2 norm within the same framework.



(a) Time vs. $|\Omega|/n$ for different sizes

(b) Time vs. $|\Omega|/n$ for different ranks

(c) Time vs. noise std std

Figure 1: (a) Time (in seconds, log scale) taken for reconstruction vs. $|\Omega|/n$ for rank 5 square matrices of different sizes n by MF-LRSDP and OptSpace. MF-LRSDP takes less time. (b) Time (in seconds, log scale) taken for reconstruction vs. $|\Omega|/n$ for 500 × 500 matrices of different ranks by MF-LRSDP and OptSpace. Again, MF-LRSDP is more efficient. (c) Time (in seconds, log scale) taken for reconstruction vs. noise standard deviation for rank 5, 200 × 200 matrices by MF-LRSDP, OptSpace, alternation and damped Newton.

	MF-LRSDP	Alternation	Damped Newton	OptSpace
Dinosaur	35.5	29.3	8.2	1.4
Longer Dinosaur	91.8	145.6	39.0561	5.3376
Giraffe	120.1	55.4	1395.9	9.7
Face	46.5	122.6	33.1	8.2

Table 1: Average time (in seconds) taken for reconstruction of the four real sequences. OptSpace takes the least time but its reconstruction results are poor. Other algorithms give comparable results except for the Giraffe sequence where the damped Newton takes a long time to converge. This is because damped Newton is not efficient for (moderate-sized) square matrices.

1.2 Evaluation with Real Data

Affine SfM. We test the performance of MF-LRSDP, alternation, damped Newton and OptSpace on a 'longer Dinosaur' [1] sequence for which $\mathbf{M}$ is a 72 × 1441 matrix. Figure 2 shows the cumulative histogram (of 25 trials) over the RMS pixel error. MF-LRSDP gives the best performance followed by damped Newton, alternation and OptSpace.

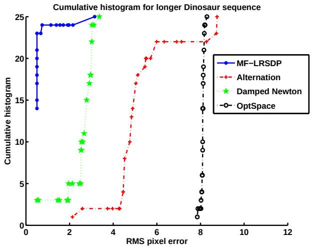


Figure 2: Cumulative histogram (of 25 trials) for the longer Dinosaur sequence. MF-LRSDP gives the best performance followed by damped Newton, alternation and OptSpace.

Timing Evaluations of Real Data. Table 1 shows the average time taken by the algorithms on the different sequences. We note that OptSpace is the most efficient algorithm, but it's poor accuracy makes it unsuitable for our problems. Other algorithms give comparable timing result except for the Giraffe sequence where the damped Newton takes a long time to converge. This is because damped Newton is not efficient for moderate-sized square matrices. On the other hand, it is quite efficient for large tall or fat matrices (where either m or n is small) as can be seen for the face dataset, where M is a 2944 × 20 matrix. This can be attributed to it's 'reduced' implementation [1].

---

[1] http://www.robots.ox.ac.uk/ vgg/data/data-mview.html

References

[1] A. M. Buchanan and A. W. Fitzgibbon. Damped newton algorithms for matrix factorization with missing data. In *CVPR*, 2005.

[2] K. Mitra, S. Sheorey, and R. Chellappa. Large-scale matrix factorization with missing data under additional constraints. In *NIPS*, 2010.