Prototyping Recommender Systems using Stochastic Average Gradient

James Lo
Computer Science
University of British Columbia
Vancouver, Canada
tklo@cs.ubc.ca

ABSTRACT

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Information Filtering*

General Terms

Asymptotic Time Complexity, Asymptotic Space Complexity, Experimentation

Keywords

Recommender systems, collaborative filtering, matrix factorization, stochastic gradient

1. INTRODUCTION

Shopping, text advertising, display advertising, renting movies, listening to music... recommender systems are prevalent and ubiquitous in our daily lives. Matrix factorization (*MF*) is a popular technique in model-based recommender systems. *MF* has been utilized extensively in past research for handling both explicit [4, 9, 7] ratings, and implicit [1, 2, 8, 3, 9] feedback.

In recommender systems that utilize matrix factorization, most optimize an objective function. Full deterministic gradient (*FG*) and stochastic gradient (*SG*) are the two main gradient methods for optimization. All of the recommender systems that we cited above utilize full deterministic gradient, or stochastic gradient.

Unfortunately, both full deterministic gradient and stochastic gradient have pitfalls when it comes to prototyping recommender systems. Full deterministic gradient can offer high quality optimizations. However, *FG* is slow because in each iteration of optimization, *FG* has to pass through all the samples in the dataset. Stochastic gradient is relatively fast; its iteration cost is low because each iteration of *SG* looks at only one or a few samples. However, the trade-off with stochastic gradient is that it often provides relatively low quality optimizations.

High quality optimizations with a low iteration cost is important when building recommender systems. The first reason is that data scientists often have to run repeated experiments: e.g. with different objective functions, different metrics, different datasets, and different optimization parameters. The second reason is that product life cycles are shortening in the age of agile software engineering. Thus data scientists are often faced with the challenge of running more experiments and producing high quality results with less time.

In this paper, we study the challenge from the perspective of convex-optimization. We propose using the stochastic average gradient (*SAG*) method [6, 5] as a viable alternative to using *FG* and *SG* during the prototyping process. *SAG* has the distinctive advantage that its optimization quality is proven to be much better than *SG*; at the same time *SAG*'s iteration cost is asymptotically identical to *SG*. However, applying and adapting *SAG* to matrix factorization is not trivial because *SAG* requires previously-computed gradients and storing these gradients can lead to very high asymptotic space complexity. We explore the challenge with space-complexity, and addresses it by proposing a re-computation approach (SAG-RECOMPUTED) that re-computes the previously-computed gradients. SAG-RECOMPUTED preserves the low iteration cost of *SAG*. The asymptotic space complexity of our SAG-RECOMPUTED approach is as efficient as the memory-less full deterministic gradient, and stochastic gradient.

To the best of our knowledge, we are the first to

- Identify the pitfalls associated with using full deterministic gradient and stochastic gradient when data-scientists prototype model-based recommender systems.
- Propose Stochastic Average Gradient (*SAG*) as a viable alternative for yielding higher quality approximations and low iteration costs in matrix factorization.
- Extend *SAG* into SAG-RECOMPUTED for matrix factorization, resolve the space complexity challenge in adapting *SAG* from the domain of large-scale supervised-machine-learning into the domain of prototyping recommender algorithms.
- Prove in theory, that our SAG-RECOMPUTED approach has identical asymptotic time complexity and identical asymptotic space complexity as the fast and memory-less stochastic gradient method.

- Extensively evaluate and compare SAG-RECOMPUTED across multiple RecSys objective functions and diverse datasets.
- Demonstrate in practice that, even without any optimization on the implementation, SAG-RECOMPUTED still yields faster convergence despite the additional time of re-computation, and that SAG-RECOMPUTED uses memory at a level similar to the memory-less stochastic gradient.

2. BACKGROUND AND TERMINOLOGY

To motivate our paper and the space complexity challenge, we first introduce the background and the terminology that we use.

Matrix Factorization Model-based recommender systems approximate the *user-item* matrix A through the dot-product of the *user*-matrix U and the *item*-matrix V: $\hat{A} = U * V$.

The *user-item* matrix A is a *nRows*-by-*nCols* matrix. Similarly, the approximation matrix $\hat{A}$ also has *nRows* rows, and *nCols* columns.

The *user* matrix U is *nRows*-by-*nDims*: U has *nRows* rows, and *nDims* columns. *nDims* is the number of latent dimensions. The *item* matrix V is *nDims*-by-*nCols*.

Optimizing an Objective Function When trying to find a suitable *user* matrix U and a suitable *item* matrix V, many model-based recommender systems optimize an objective function with respect to U and V:

$$\arg\min_{U,V}(\text{or} \arg\max) \left[f(U,V) = \sum_{i=1}^{nRows} \sum_{j=1}^{nCols} f(\bar{u}_i, \bar{v}_j) \right]$$

When we take the gradient of the objective function with respect to a row in U (e.g. $\bar{u}_i$), we sum up the gradient of all the entries in $\hat{A}$ that belong to $\bar{u}_i$.

3. FUTURE WORK & CONCLUSION

This paper is the first in the series of our study on data scientists prototyping model-based recommender systems. We explored the convex-optimization perspective of the problem; and we propose Stochastic Average Gradient as a viable alternative to Full Deterministic Gradient and Stochastic Gradient. In theory, we proved that our extension and adaptation of *SAG* has asymptotic time complexity and asymptotic space complexity as efficient as stochastic gradient. In practice, through extensive evaluation we demonstrated that, even without any fine-tuning or optimization of the implementation, SAG-RECOMPUTED still outperforms both Full Deterministic Gradient and Stochastic Gradient in terms of faster convergence.

In the future, we also aim to complete our ongoing work on the metrics perspective and on the software engineering perspective. Given a dataset, the quality of a recommender system is often evaluated in various metrics: e.g. precision, recall, area under curve, reciprocal rank, NDCG, and variants of the above such as top-K precision and top-K hit rate. Many papers in the literature claim their objective function is better by illustrating that their objective function performs in some of these metrics better than other objective functions. Therefore, in the metrics perspective, we are exploring and investigating which factors are more relevant and important towards scoring high in the various metrics: is it the objective function, the method for convex-optimization such as *SAG*, other fine-tuning mechanisms such as bootstrapping, or the hyper parameters that we use in convex-optimization but which can also be specific to the dataset.

In the software engineering perspective, we are studying how to increase the productivity of data scientists. At this point, we are designing and developing a *mix-n-match* or *plug-n-play* framework that enables data scientists to very rapidly prototype and experiment different combinations of objective functions, datasets, gradient methods, hyper parameters and evaluation metrics.

4. REFERENCES

[1] Y. Hu, Y. Koren, and C. Volinsky. Collaborative filtering for implicit feedback datasets. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, pages 263–272. IEEE, 2008.

[2] R. Pan, Y. Zhou, B. Cao, N. N. Liu, R. Lukose, M. Scholz, and Q. Yang. One-class collaborative filtering. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, pages 502–511. IEEE, 2008.

[3] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pages 452–461. AUAI Press, 2009.

[4] J. D. Rennie and N. Srebro. Fast maximum margin matrix factorization for collaborative prediction. In *Proceedings of the 22nd international conference on Machine learning*, pages 713–719. ACM, 2005.

[5] N. L. Roux, M. Schmidt, and F. R. Bach. A stochastic gradient method with an exponential convergence _rate for finite training sets. In *Advances in Neural Information Processing Systems*, pages 2663–2671, 2012.

[6] M. Schmidt, N. L. Roux, and F. Bach. Minimizing finite sums with the stochastic average gradient. *arXiv preprint arXiv:1309.2388*, 2013.

[7] Y. Shi, A. Karatzoglou, L. Baltrunas, M. Larson, and A. Hanjalic. Gapfm: Optimal top-n recommendations for graded relevance domains. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, pages 2261–2266. ACM, 2013.

[8] Y. Shi, A. Karatzoglou, L. Baltrunas, M. Larson, N. Oliver, and A. Hanjalic. Climf: learning to maximize reciprocal rank with collaborative less-is-more filtering. In *Proceedings of the sixth ACM conference on Recommender systems*, pages 139–146. ACM, 2012.

[9] H. Steck. Training and testing of recommender systems on data missing not at random. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 713–722. ACM, 2010.