Review of SLIM Optimization via ADMM

Hsin Hsien Chung
hhc3@illinois.edu

INTRODUCTION

The Sparse Linear Method (SLIM) is one of the best approaches in terms of recall and hit-rate for top-N recommendations [2]. It requires less computing time than model-base matrix factorization and has greater accuracy than the item-based and user-based collaborative filtering. However, it is still computationally expensive to train the large data set while the recommendation systems are heavily used in eCommerce, social media and video streaming, where the recommendation should be pushed to users in a timely manner. In the paper, an optimization of the SLIM is using the Alternating Directions Method of Multipliers (ADMM) with augmented Lagrangian.

MODEL AND OPTIMIZATION

The SLIM model [2] is defined as follows. In Equation 1, the score of the user u and item j is defined as $S_{u,j}$, where X is the user-item matrix and B is the sparse aggregation

$$S_{u,j} = X_{u,\cdot} \cdot B_{\cdot,j} \tag{1}$$

coefficient matrix learned by solving L_1 -norm and L_F -norm regularized optimization problem in Equation 2. The following is the equation of the coefficient matrix B. The L_1 -norm regularization preserved the sparsity where the L_F -norm measures model complexity and

$$\begin{aligned} & min_B \ \tfrac{1}{2} \cdot \|X - XB\|_F^2 + \tfrac{\lambda_2}{2} \cdot \|B\|_F^2 + \lambda_1 \cdot \|B\|_1 \\ & s.t. \ \ diag(B) = 0 \\ & \qquad B_{i,j} \geq 0 \ \ \forall i, j \in I \end{aligned} \tag{2}$$

prevents overfitting. The equation is subject to that the diagonal of B has to be 0 to avoid trivial solutions, preventing calculating the item score using itself, and also the weight should be non-negative.

In this paper [1], the Alternating Directions Method of Multipliers (ADMM) is used to optimize the coefficient matrix B. The ADMM equation is defined in Equation 3, where $f(B)$

$$\begin{aligned} min_{B,C} &\quad f(B) + g(C) \\ s.t. &\quad B = C \end{aligned} \tag{3}$$

and $g(C)$ are defined in Equation 4 and 5. By adding the L_1 penalty term in Equation 6, the equality constraint $B = C$ is absorbed into the augmented Lagrangian. The ADMM will

$$f(B) = \frac{1}{2}\|X - XB\|_F^2 + \frac{\lambda_2}{2} \cdot \|B\|_F^2, \ dom\ f = \{B \mid diag(B) = 0\} \tag{4}$$

$$g(C) = \lambda_1 \cdot \|C\|_1 \qquad\qquad , \ dom\ g = \{C \mid C_{i,j} \geq 0\} \tag{5}$$

$$L_\varrho(B, C, \Gamma) = f(B) + g(C) + \langle \Gamma, B - C \rangle_F + \frac{\varrho}{2} \cdot \|B - C\|_F^2 \tag{6}$$

iterate the updated equation in Equation 7 and reach to an optimal coefficient matrix B after convergence or usually 50 iterations [1].

$$\begin{aligned} B^{(k+1)} &= argmin_B\, L_\varrho(B, C^{(k)}, \Gamma^{(k)}) \\ C^{(k+1)} &= argmin_B\, L_\varrho(B^{(k+1)}, C, \Gamma^{(k)}) \\ \Gamma^{(k+1)} &= \Gamma^{(k)} + \varrho \cdot (B^{(k+1)} - C^{(k+1)}) \end{aligned} \tag{7}$$

ANALYSIS

One of the advantages of using ADMM is its flexibility of switching on and off the constraints and regularization terms in the original SLIM-objective. There are several models compared in this paper [1] using the 3 data-set with different sizes. There are three groups of ADMM models. First is the original ADMM model including the non-negative weights (ADMM ≥ 0), L_1 -norm and L_2 -norm regularization (ADMM L_1), and both non-negative weights and L_1 -norm and L_2 -norm regularization (ADMM ≥ 0 & L_1). The second one is the dense-weight matrices (Dense and Dense ≥ 0), and sparse-weight matrices (Sparse Approx. and Sparse Approx. ≥ 0). These are closed form solutions derived from dropping L_1 penalty. The third variant is the model trained on centered data using item-specifics (Centered, Centered & item-L_2 , and Centered & item-L_2 & L_1).

In general, the solutions with non-negative weights have better accuracy in large data sets. The Dense and Sparse Approx. preference similar to ADMM L_1 , yet the Sparse solution significantly reduces the training time. The Centered with more constraints yield to the better accuracy. In addition, all the computation time is much faster than the original SLIM and other models.

Interestingly, when comparing the ADMM L_1 with Dense and Sparse Approx., ADMM L_1 seems to perform better in smaller sample sizes, and perform even better in the non-negative weight case.

FURTHER WORK

In the paper [1], it only compares the original SLIM method instead of other SLIM variations. The feature selection method applied to the original SLIM method has a better result compared to the origin SLIM method. However, this scenario is not addressed in the paper. With the feature selection, it can substantially lower the SLIM computational requirements and learning time with minimal quality degradation [2]. In addiction, it should be beneficial and easy to analyze if the computation time for each method is listed.

CONCLUSION

The idea of applying ADMM to optimize SLIM is a relatively new way to optimize SLIM. There should be more research on it with different analysis. However, it is clear that the application improves recommendation accuracy with less computing time in different variant solutions. For the original ADMM, it works well in smaller datasets which can be used in cold-start data sets. The L_1 penalty does not improve accuracy, and if it is dropped, the ADMM is not necessary, and the closed form solutions Dense and Sparse Approx. can be used with less training time. The L_2 -norm regularization in centered item-specific models does not decrease the training time, yet it results in greater accuracy. The properties can be switched on and off easily, so it can produce different results based on the focus of the application.

REFERENCES

[1] H. Steck, M. Dimakopoulou, N. Riabov, and T. Jebara. 2020. ADMM SLIM: Sparse Recommendations for Many Users. *WSDM '20: Proceedings of the 13th International Conference on Web Search and Data Mining*. 555-563.

[2] X. Ning and G. Karypis. 2011. SLIM: Sparse Linear Methods for Top-N Recommender Systems. *IEEE International Conference on Data Mining (ICDM)*. 497-506.

[3] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. 2011. *Foundations and Trends in Machine Learning*, 3(1):1–122.