Algorithms Associated with Factorization Machines

Yanyu Liang
Computational Biology Department
Carnegie Mellon University
Pittsburgh, PA 15213
yanyul@andrew.cmu.edu

Xupeng Tong
Computational Biology Department
Carnegie Mellon University
Pittsburgh, PA 15213
xtong@andrew.cmu.edu

Xin Lu
Computational Biology Department
Carnegie Mellon University
Pittsburgh, PA 15213
xlu2@andrew.cmu.edu

1 Introduction

Factorization machines proposed in Rendle (2010), gives a general way to map data $X \in \mathbb{R}^n$ to real value Y in a target domain $C \subseteq \mathbb{R}$ as follow (for simplicity, only second-order interaction is considered here):

$$\begin{aligned}\hat{y}(\vec{x}) &= w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} \langle \vec{v}_i, \vec{v}_j \rangle x_i x_j \\ &= w_0 + \vec{w}^T \vec{x} + \vec{x}^T V V^T \vec{x} \quad \text{, where } V \in \mathbb{R}^{p \times k}\end{aligned}$$

In practice, with suitable loss function, FMs can be used for classification, regression, and ranking. FMs has a well established learning algorithm that have been successfully applied in many areas, is specially designed for problem that involves sparse input (e.g recommender system). By introducing factorized parametrization (or in another sense that to add additional model complexity constraint), factorization machines are reliable in the setting that data vector are almost zero. Besides, FMs have linear time complexity and they can easily mimic many different flavors of factorization models like matrix factorization, parallel factor analysis or specialized models like SVD++, PITF or FPMC by feature engineering. However, problems still exist for the original formulation of factorization machines: i) FMs introduce a non-convex optimization with no global minimum guarantee; ii) the rank of parameter matrix needs to be set as a hyper-parameter, which requires a huge amount of computing in model selection; iii) it has no sparsity constraint on parameter matrix, which may restrict its application outside matrix completion.

To tackle the non-convex issue, Yamada et al. (2015) used symmetric parameter matrix Z instead of the original form VV^T and train the symmetric parameter matrix Z directly (called convexFMs). Besides, they replace the original squared Frobenius matrix norm with a nuclear norm (trace norm) in the objective function. The original problem, after all these transformations, is casted into a convex optimization problem where the global optimal could be achieved. Also, the new formulation reduces the restrictions on feature interaction weight matrix Z compared to the vanilla version, e.g Z is not required to be positive semi-definiteness. What's more, the issue of sparse inputs is tackled by penalizing the high rank of matrix parameter Z, which is controlled by β and not dependent on the hyper-parameter k (matrix rank).

To solve the formulated convex optimization problem, two-block coordinate descent algorithm is used Yamada et al. (2015). By dividing the original objective function into two, the feature weight vector and the low-rank feature interaction weight matrix Z could be solved separately. Further, in order to estimate the value of $\vec{w}$, standard coordinate descent is directly applied, while the estimation of Z utilizes a greedy coordinate descent algorithm, where the eigendecomposition of Z is maintained. And recently, Lin and Ye (2016) proposed a generalized version of FMs, gFMs, by relaxing VV^T to UV^T, and they showed that with $O(\epsilon)$ reconstruction error, the sampling complexity needed is $O(k^3 p \log(1/\epsilon))$. And they also proposed an algorithm that converges linearly implemented in Python.

Furthermore, to introduce sparsity to Z, we need to add a constraint saying that every row in V should be orthogonal to each other. In 2014, Vervier et al. (2014) proposed a matrix penalty term that could do this. Therefore, it is straightforward to extend the original FMs by introducing such orthogonal constraint. Alternatively, with Lasso penalty in hand, the sparsity can also be introduced with $\|\text{vec}(VV^T)\|_1$.

2 Our Plan

For solving convexFMs, there are several ways of addressing the nuclear norm regularization problems and one out of many is to to use proximal gradient descent, which commonly involves the proximal operator and iterative soft thresholding on singular values Cai et al. (2010), is time consuming when scaling to large matrix sizes. However, through years of study, many algorithms have been developed to address the scalability of nuclear norm problem like Tan et al. (2016) using method on the modification of proximal gradient descent. So, before second milestone, we plan to implement an efficient proximal descent method on convexFMs, and compare the proximal method with the coordinate descent one proposed in Yamada et al. (2015). For gFMs, we plan to implement the algorithm posted in Lin and Ye (2016) using TensorFlow before second milestone, And after that, we will compare it performance with algorithm provided by libFM Rendle (2012) (it has gradient descent, alternating least-squares, and MCMC).

For solving the FMs with additional sparsity constraints, we plan to first derive the basic subgradient descent method for both the orthogonal column penalty and the revised Lasso penalty before second milestone. If time permits, we will also explore the possibility to use more advanced method to solve them, such as proximal descent. Also, we will obtain some real world data from previous work on FMs and generate some simulated data for testing the performance of our algorithm in different cases.

Also, we may explore the possibility to extend the FMs to solve tensor factorization problem.

References

S. Rendle, "Factorization machines," in *2010 IEEE International Conference on Data Mining*, Dec 2010, pp. 995–1000.

M. Yamada, A. Goyal, and Y. Chang, "Convex factorization machine for regression," *arXiv preprint arXiv:1507.01073*, 2015.

M. Lin and J. Ye, "A non-convex one-pass framework for generalized factorization machines and rank-one matrix sensing," *arXiv preprint arXiv:1608.05995*, 2016.

K. Vervier, P. Mahé, A. D'Aspremont, J.-B. Veyrieras, and J.-P. Vert, "On learning matrices with orthogonal columns or disjoint supports," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2014, pp. 274–289.

J.-F. Cai, E. J. Candès, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM Journal on Optimization*, vol. 20, no. 4, pp. 1956–1982, 2010.

M. Tan, S. Xiao, J. Gao, D. Xu, A. van den Hengel, and Q. Shi, "Proximal riemannian pursuit for large-scale trace-norm minimization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 5877–5886.

S. Rendle, "Factorization machines with libfm," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 3, no. 3, p. 57, 2012.