Generalized Linear Models in Collaborative Filtering

Hao Wu WUHAO@STANFORD.EDU
CME 323, Spring 2016

Abstract

This study presents a distributed implementation of the collaborative filtering method based on generalized linear models. The algorithm is based on alternate convex search and presents efficient scaling with respect to the number of users. However, it is not parallel in terms of the the number of items. A sample implementation focusing on alternating logistic regression is available on GitHub (Wu, 2016).

1. Introduction

1.1. Generalized linear models

Generalized linear models (GLM) is a generalization of the ordinary linear regression from the normal distribution to other distributions in the exponential families (McCullagh & Nelder, 1989). The family of GLM includes many popular linear models for regression analysis such as linear regression, logistic regression and Poisson regression.

The basic GLM features two essential model components:

- A probability distribution from the exponential families.
- A linear parameter model

The density function of a distribution in the exponential families in the canonical form can be expressed as

$$f_Y(y|\boldsymbol{\eta}) = h(y)\exp(\boldsymbol{\eta}^T\mathbf{T}(y) - \mathbf{A}(\boldsymbol{\eta}))\,, \tag{1}$$

where $\boldsymbol{\eta}$ is the natural parameters, $\mathbf{T}(y)$ is the natural sufficient statistic of the family, and $\mathbf{A}(\boldsymbol{\eta})$ is the cumulant function. The cumulant function is closely related to the moments of $\mathbf{T}(y)$, e.g.

$$\mathbb{E}[\mathbf{T}(y)] = \frac{\partial \mathbf{A}}{\partial \boldsymbol{\eta}^T}\,, \tag{2}$$

$$\mathrm{Var}[\mathbf{T}(y)] = \frac{\partial^2 \mathbf{A}}{\partial \boldsymbol{\eta}\partial \boldsymbol{\eta}^T}\,. \tag{3}$$

As Eq. 2 connects the expectation of $\mathbf{T}(y)$ to $\boldsymbol{\eta}$, it is conventionaly referred to as the link function in the context of GLM and is often denoted by $g^{-1}(\boldsymbol{\eta})$.

GLM assumes the natural parameter $\boldsymbol{\eta}$ to be a linear function of the predictor $\mathbf{X}$, i.e.

$$\boldsymbol{\eta} = \mathbf{X}\boldsymbol{\beta}\,. \tag{4}$$

Therefore, the log-likelihood function for independent samples is in the from

$$\log \mathcal{L}(\mathbf{y}|\mathbf{X},\,\boldsymbol{\beta}) = \sum_{i=0}^{N} \mathbf{x}_i^T\boldsymbol{\beta}\mathbf{T}(y_i) - \mathbf{A}(\boldsymbol{\beta}^T\mathbf{x}_i) - \log\left(h(y_i)\right)\,. \tag{5}$$

Given property in Eq.3 and positive definiteness of the covariance matrix, the log-likelihood function is strictly convex with respect to $\boldsymbol{\beta}$ if the family is minimal.

Note that $\mathbf{x}_i^T\boldsymbol{\beta}$ involves the data only via the sufficient statistic $\mathbf{T}$, which will become handy for data reduction in latter sections. In addition, distributions in the exponential families have conjugate priors, which is convenient for Bayesian statistics.

As the exponential families include normal distribution, multinomial distribution, and poisson distribution, GLM unifies linear regression, logistic regression and Poisson regression, etc.

1.2. Collaborative Filtering based on alternating least squares

Collaborative filtering (CF) is a technique used by some recommender systems aiming to estimate an user-item association matrix $\mathbf{H}$. The $n_u \times n_m$ matrix $\mathbf{H}$ is often assumed to be of rank k. Thus, it can be written in the factorized from of $\mathbf{H} = \mathbf{U}\mathbf{M}^T$, where $\mathbf{U}$ is an $n_u \times k$ user-feature matrix and $\mathbf{M}$ is an $n_m \times k$ item-feature matrix.

Prepared for the final project of CME 323 in Spring 2016.

The alternating least squares (ALS) based CF Alg. (Koren et al., 2009) attempts to solve the following minimization problem

$$\min_{\mathbf{U},\,\mathbf{M}} \sum_{(u,\,m)\in\kappa} (y_{um} - \mathbf{u}_u^T \mathbf{m}_m)^2 \,, \tag{6}$$

where y_{um} is the series of training data representing the rating of user u of item m and κ is the set of the $(u,\,m)$ pairs for which y_{um} is known.

This formulation leads to a biconvex optimization problem, which can be solved by alternate convex search (ASC). It alternatively updates $\mathbf{U}$ and $\mathbf{M}$ while fixing the other. The update procedure is the same as solving a series of ordinary least-square problems. Under certain regularity conditions, the sequence generated by ACS converges monotonically. The so obtained solution is a partial optimum, but it neither has to be a global nor a local optimum to the given biconvex optimization problem, as stationary points can be saddle points of the given function (Gorski et al., 2007).

2. Generalized linear models in collaborative filtering

2.1. Maximum likelihood estimation

The CF model based on the ALS algorithm in Sec. 1.2 can also be formulated in the context of GLM. If the rating data are independently sample from a normal distribution whose mean is of low rank, i.e.

$$y_i \overset{iid}{\sim} \mathcal{N}(\mathbf{u}_{u_i}^T \mathbf{m}_{m_i},\, \sigma^2) \,, \tag{7}$$

where u and m are the user and item indices associated with y_i, we will have the maximum likelihood estimation in the form of

$$\min_{\mathbf{U},\,\mathbf{M}} \sum_{i=0}^{N} (y_i - \mathbf{u}_{u_i}^T \mathbf{m}_{m_i})^2 \,. \tag{8}$$

where u_i and m_i denotes the user and item index of the i-th entry. Other than allowing multiple entries for a certain $(u,\,m)$ pair, this formulation is essentially the same as Eq. 6.

More generally, if we have the data independently sampled from a distribution of the exponential family whose natural parameter is of low rank, i.e.

$$y_{u_i m_i} \overset{iid}{\sim} \mathcal{D}(\boldsymbol{\eta}_{um} = \mathbf{u}_{u_i}^T \mathbf{m}_{m_i}) \,, \tag{9}$$

the maximum likelihood estimation can be obtained in the form

$$\min_{\mathbf{U},\,\mathbf{M}} \sum_{u,\,m} n_{um} \Big[-\mathbf{u}_u^T \mathbf{m}_m \mathbf{T}_{um} + \mathbf{A}(\mathbf{u}_u^T \mathbf{m}_m) \Big] \,, \tag{10}$$

where n_{um} are the number of entries generated by user u on item m and

$$\mathbf{T}_{um} = \frac{1}{n_{um}} \sum_{(u_i,\,m_i)=(u,\,m)} \mathbf{T}(y_i) \tag{11}$$

is the corresponding sufficient statistic summarizing these entries. Therefore, the total number of function evaluations during the optimization process is determined by the number of distinct $(u,\,m)$ pairs instead of number of entries, which can potentially be orders of magnitude larger depending on the application scenario.

Again, the convexity of GLM ensures that Eq. 10 is biconvex. The ASC to this problem involves series of GLM regressions, whose implementation is likely to be readily available.

2.2. Regularization

The most common regularization for the CF model based on the ALS algorithm is the sum of Frobenius norms,

$$R_*(\mathbf{U},\,\mathbf{M}) = \frac{1}{2}\big(\|\mathbf{U}\|_F + \|\mathbf{M}\|_F\big) \,. \tag{12}$$

The regularization can be interpreted as a normal prior for $\mathbf{U}$ and $\mathbf{M}$. More importantly, it can be related to the nuclear norm of $\mathbf{H}$ (Srebro et al., 2004; Mazumder et al., 2010) as

$$\|\mathbf{H}\|_* = \min_{\mathbf{U},\,\mathbf{M}:\,\mathbf{H}=\mathbf{U}\mathbf{M}^T} \frac{1}{2}\big(\|\mathbf{U}\|_F + \|\mathbf{M}\|_F\big) \,. \tag{13}$$

Therefore, it is equivalent to using $\|\mathbf{H}\|_*$ as the regularization term (Hastie et al., 2014). Moreover, this form of regularization also eliminates the arbitrary scaling between $\mathbf{U}$ and $\mathbf{M}$.

Additionally, if $\mathbf{U}$ and $\mathbf{M}$ are sparse, we can further regularize the problem by the element-wise L1 norm,

$$R_1(\mathbf{U},\,\mathbf{M}) = \|\mathbf{U}\|_1 + \|\mathbf{M}\|_1 \,. \tag{14}$$

The combination of these two forms of the regularization gives the matrix form of the elastic-net regularization (Zou & Hastie, 2005). With this, the problem in Eq. 10 is modified to be

$$\begin{aligned} \min_{\mathbf{U},\,\mathbf{M}} \sum_{u,\,m} n_{um} \Big[&-\mathbf{u}_u^T \mathbf{m}_m \mathbf{T}_{um} + \mathbf{A}(\mathbf{u}_u^T \mathbf{m}_m) \Big] \\ &+ \lambda \Big[\alpha\big(\|\mathbf{U}\|_1 + \|\mathbf{M}\|_1\big) \\ &+ (1-\alpha)\frac{1}{2}\big(\|\mathbf{U}\|_F + \|\mathbf{M}\|_F\big) \Big] \,. \end{aligned} \tag{15}$$

The ASC to Eq. 15 then becomes a series of elastic-net regularized GLM regression, which is again commonly available.

2.3. Example: alternating logistic regression

One of the important application of collaborative filtering is to predict the user behavior based on indirect feedback. These feedback usually of the binary type. For instance, online-advertising agency may be interested in predicting if a user will click a certain (category of) banner ad(s) based on the past user-ad interaction data which records the click/no-click history. The probability of the occurrence of clicking given the impression is called the click-through rate (CTR).

The clicking behavior of a user u on a certain (category of) item(s) m is assumed to follow the Bernoulli distribution with a fixed probability $p_{um} = \Pr(y_i = 1|(u_i, m_i) = (u, m))$. As the Bernoulli distributions form an exponential family, we can apply the method introduced in Sec. 2.1, with the natural parameter

$$\begin{aligned} \boldsymbol{\eta}_{um} &= \log \frac{p_{um}}{1 - p_{um}} \\ &= \mathbf{u}_u^T \mathbf{m}_m \end{aligned} \tag{16}$$

and the regularized maximum likelihood estimation can be formularized as

$$\begin{aligned} \min_{\mathbf{U},\mathbf{M}} \sum_{u,m} n_{um} \Big[&- \mathbf{u}_u^T \mathbf{m}_m \mathbf{T}_{um} + \log\big(1 + \exp(\mathbf{u}_u^T \mathbf{m}_m)\big) \Big] \\ &+ \lambda \Big[\alpha \big(\|\mathbf{U}\|_1 + \|\mathbf{M}\|_1 \big) \\ &+ (1 - \alpha) \frac{1}{2} \big(\|\mathbf{U}\|_F + \|\mathbf{M}\|_F \big) \Big], \end{aligned} \tag{17}$$

where

$$\mathbf{T}_{um} = \frac{1}{n_{um}} \sum_{(u_i, m_i) = (u, m)} y_i \tag{18}$$

counts the number of positive entries for a given (u, m) pair. Consequently, we can use alternating logistic regression (ALR) to solve this problem. Details of the solution procedure will be discussed in Sec. 3.

3. Distributed algorithm

3.1. Description

In this section, a distributed algorithm for the alternating logistic regression problem is discussed. The algorithm is developed in the Spark framework. Key features of the algorithm include

- efficient scaling with respect to the number of users,
- extensive reuse of the available GLM implementation,
- extendable to other types of distribution of the exponential family.

Algorithm 1 Distributed alternating logistic regression

Input: data: $\mathbf{RDD}\big[(\mathbf{u_i}, \mathbf{m_i}), \mathbf{y_i}\big]$
Form sufficient statistic T: $\mathbf{RDD}\big[(\mathbf{u}, \mathbf{map}(\mathbf{m}, \mathbf{y}))\big]$
Initialize U: $\mathbf{RDD}\big[(\mathbf{u}, \mathbf{U_u})\big]$, M: $\mathbf{Array}\big[\mathbf{M_m}\big]$
repeat
  D = T.join(U) (co-partitioned)
  Update M
  Broadcast M
  Upate U
until converge

A typical user-item association matrix is assumed, where the number of users is much greater than the number of items, which is itself much greater than the rank, $n_u \gg n_m \gg k$. We further assume that $n_m \times k$ is small enough to fit inside a single machine.

The outline of the algorithm is summarized in Alg. 1. The input data is in the format of $\mathbf{RDD}\big[(\mathbf{u_i}, \mathbf{m_i}), \mathbf{y_i}\big]$. This is first transformed into $\mathbf{RDD}\big[(\mathbf{u}, \mathbf{map}(\mathbf{m}, \mathbf{y}))\big]$, which represents the sufficient statistic in Eq. 11. This step is achieved by reduceByKey and aggregateByKey. It reduces the number of the entries while preserving the sparsity pattern.

The initial guesses of U and M are generated randomly. The user feature matrix U is and $\mathbf{RDD}\big[(\mathbf{u}, \mathbf{vector})\big]$ co-partitioned as T. The item feature matrix M is a local array of vectors.

For each iteration, a co-partitioned join between U and T is performed. With this, the item feature matrix M is first updated, followed by broadcasting it to each machine. The user feature matrix U is updated at the end of the iteration.

The update of item feature matrix M utilizes the distributed regression model in spark.ml. It consists of n_m serial calls of the regression training method. This procedure is shown in Alg. 2.

The update of user feature matrix U (Alg. 3) consists of n_u parallel execution of local regression models. This can be easily achieved via mapping. The implementation of the local regression models are modified from the distributed version in spark.ml, by changing the format of the input data from DataFrame to regular Scala array. The optimization method and other features are kept the same.

3.2. Analysis

The analysis assumes the distributed algorithm running on p machines, with n_u users, n_m items. Each user has records on at most n_l items and the total number of entries is N, which can be much greater than $n_u n_l$. The model is assumed to be of rank k.

ALL-TO-ALL: SHUFFLE SIZE

- Form T: $\mathbf{RDD}\big[(\mathbf{u},\ \mathbf{map}(\mathbf{m},\ \mathbf{y}))\big]$

  reduceByKey: $\mathcal{O}\big(\min(p \cdot n_u n_l,\ N)\big)$

  aggregateByKey: $\mathcal{O}\big(\min(p \cdot n_u n_l,\ N)\big)$

- Within iteration:

  D = T.join(U): co-partitioned, no shuffleing

ONE-TO-ALL / ALL-TO-ONE: COMMUNICATION COST

- Within iteration:

  Broadcast M: $\mathcal{O}\big(\log(p) n_m k\big)$

  n_m rounds of distributed GLM:

  copmute gradient: $\mathcal{O}\big(\log(p) n_m k\big)$

OTHER COMPUTATION COST

- Within iteration:

  n_m rounds of distributed GLM:

  L-BFGS: $\mathcal{O}\big(k n_m\big)$

  n_u parallel local GLM:

  copmute gradient: $\mathcal{O}\big(kL\big)$

  L-BFGS: $\mathcal{O}\big(k\big)$

4. Result

A sample implementation featuring the ALR method is a hosted on GitHub (Wu, 2016).

The implementation is tested using data generated by random low rank $\mathbf{H}$ matrix. Since the implementation alters the Spark library, the test is based on a locally-built Spark with multi-core single machine. A sample estimated probability using ALR is shown in Fig. 1.

The scaling of computational time with respect to N, k, n_l, n_m, n_u, and are tested. The results are shown in Fig. 2, 3, 4, 5, 6.

5. Conclusion

In conclusion, a generalization of the collaborative filtering is proposed using the concept of Generalized Linear Models is proposed. A distributed alternate convex search algorithm to the generalized collaborative filtering problems is analyzed. The algorithm presents efficient scaling with respect to the number of users but is not parallel in terms of the the number of items. A sample implementation focusing on alternating logistic regression is presented and is available on GitHub (Wu, 2016).

Algorithm 2 Update of item feature matrix M

Input: D: $\mathbf{RDD}\big[\mathbf{u}, (\mathbf{map}(\mathbf{m}, (\mathbf{T_{um}}, \mathbf{n_{um}})), \mathbf{U_u})\big]$
for $m = 0$ **to** n_m **do**
  Filter D with m to construct DataFrame df
  Distributed GLM(df)
  Update M(m)=GLM.coefficients
end for

Algorithm 3 Update of user feature matrix U

Input: D: $\mathbf{RDD}\big[\mathbf{u}, (\mathbf{map}(\mathbf{m}, (\mathbf{T_{um}}, \mathbf{n_{um}})), \mathbf{U_u})\big]$,
M: $\mathbf{Array}\big[\mathbf{M_m}\big]$
U = D.mapValues(v $\Rightarrow$ Local GLM(v, M))

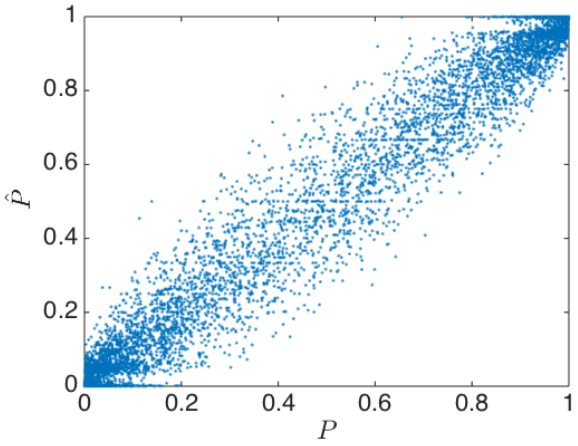

Figure 1. Scatter plot of the true probability verses the estimated probability using ALR, with $m_u = 2000$, $n_m = 5$, $k = 4$, $N = 200000$.

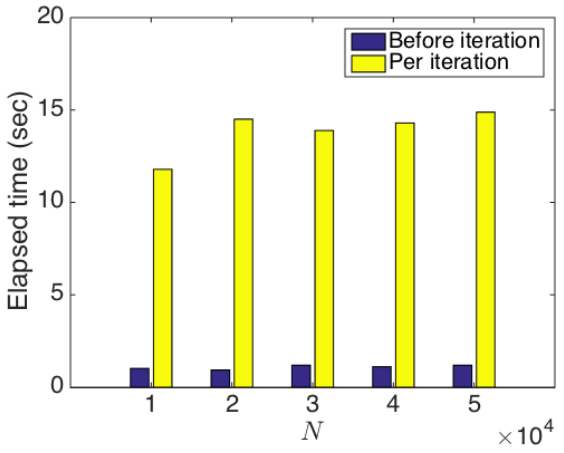

Figure 2. Scaling to elapsed time with respect to N, $k = 2$, $n_l = 10$, $m_u = 1000$, $n_m = 100$.

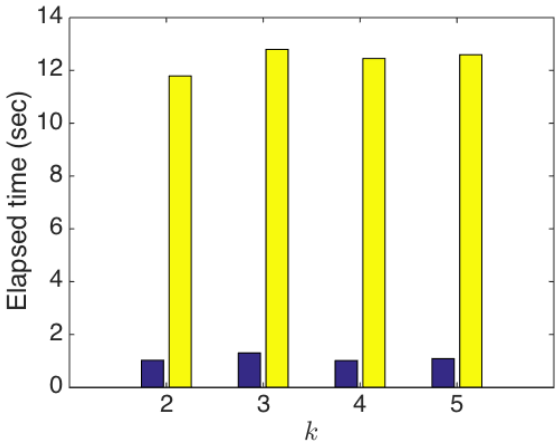

Figure 3. Scaling to elapsed time with respect to k, $N = 10000$, $n_l = 10$, $n_u = 1000$, $n_m = 100$.

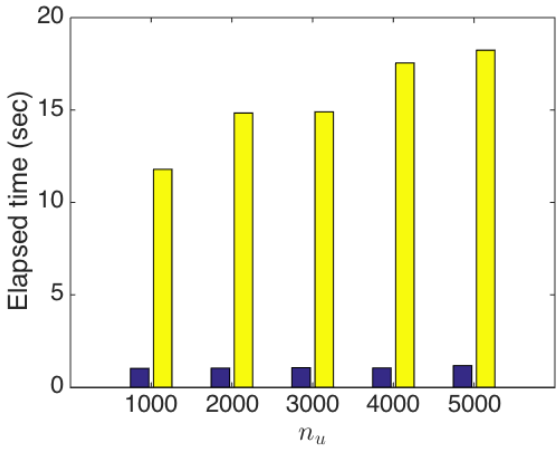

Figure 5. Scaling to elapsed time with respect to m_u, $N = 10000$, $k = 2$, $n_l = 10$, $n_m = 100$.

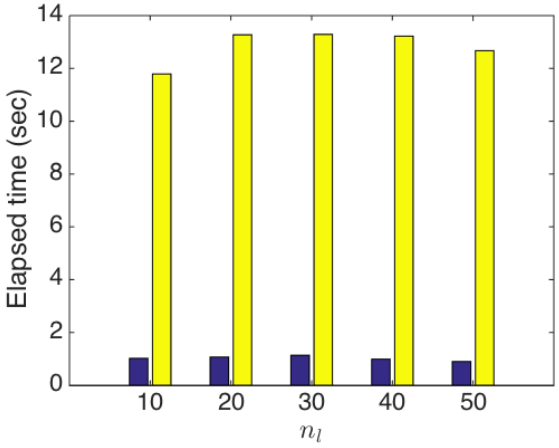

Figure 4. Scaling to elapsed time with respect to n_l, $N = 10000$, $k = 2$, $n_u = 1000$, $n_m = 100$.

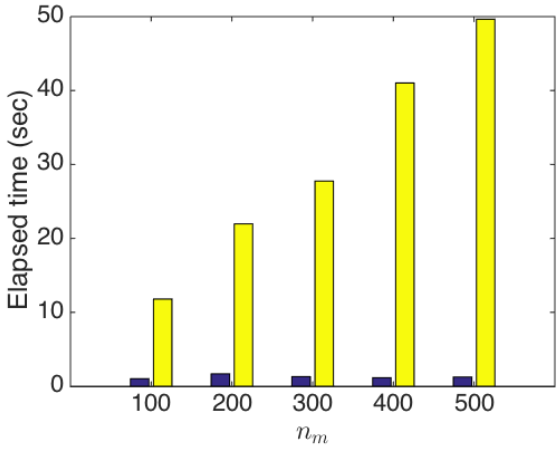

Figure 6. Scaling to elapsed time with respect to m_m, $N = 10000$, $k = 2$, $n_l = 10$, $n_u = 1000$.

References

Gorski, Jochen, Pfeuffer, Frank, and Klamroth, Kathrin. Biconvex sets and optimization with biconvex functions: a survey and extensions. *Mathematical Methods of Operations Research*, 66(3):373–407, 2007.

Hastie, Trevor, Mazumder, Rahul, Lee, Jason, and Zadeh, Reza. Matrix completion and low-rank svd via fast alternating least squares. *arXiv preprint arXiv:1410.2596*, 2014.

Koren, Yehuda, Bell, Robert, and Volinsky, Chris. Matrix factorization techniques for recommender systems. *Computer*, (8):30–37, 2009.

Mazumder, Rahul, Hastie, Trevor, and Tibshirani, Robert. Spectral regularization algorithms for learning large incomplete matrices. *The Journal of Machine Learning Research*, 11:2287–2322, 2010.

McCullagh, Peter and Nelder, John A. *Generalized linear models*, volume 37. CRC press, 1989.

Srebro, Nathan, Rennie, Jason, and Jaakkola, Tommi S. Maximum-margin matrix factorization. In *Advances in neural information processing systems*, pp. 1329–1336, 2004.

Wu, Hao. SparkALR, June 2016. URL http://dx.doi.org/10.5281/zenodo.54849.

Zou, Hui and Hastie, Trevor. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2): 301–320, 2005.