Supplementary Materials for "LIBMF: A Library for Parallel Matrix Factorization in Shared-memory Systems"

Wei-Sheng Chin, Bo-Wen Yuan, Meng-Yuan Yang, Yong Zhuang, Yu-Chin Juan, Chih-Jen Lin

Department of Computer Science, National Taiwan University, Taipei 10617, Taiwan

1. The Parallel Stochastic Gradient Method in LIBMF

In LIBMF, we implement a parallel stochastic gradient method to solve RVMF, BMF, as well as OCMF under the framework proposed by Chin et al. (2015a). First, the training matrix is divided into many blocks and some working threads are created. Then, a block scheduler assigns independent blocks to threads that run stochastic gradient method (SG) in parallel. We use the learning rate scheduler in Chin et al. (2015b) and further extend it to cover more loss and regularization functions by following Duchi et al. (2011). In the rest of Section 1, we derive the SG update rules for RVMF and BMF, and the algorithm of OCMF is left in Section 2.

Let R be a training matrix and $P \in \mathbb{R}^{k\times m}$ and $Q \in \mathbb{R}^{k\times n}$ standard for the two factor matrices. Both RVMF and BMF can be formulated as an unconstrained optimization problem.

$$\min_{P,Q} \quad \sum_{(u,v)\in R} \phi_{u,v}(P,Q) + \rho_{u,v}(P,Q), \tag{1}$$

where

$$\begin{aligned}\phi_{u,v}(P,Q) &= f(\boldsymbol{p}_u, \boldsymbol{q}_v; r_{u,v}) + \lambda_p \|\boldsymbol{p}_u\|_2^2 + \lambda_q \|\boldsymbol{q}_v\|_2^2 \\ \rho_{u,v}(P,Q) &= \mu_p \|\boldsymbol{p}_u\|_1 + \mu_q \|\boldsymbol{q}_v\|_1 .\end{aligned}$$

In (1), u and v are respectively the row index and the column index of R, and $f(\cdot)$ is a non-convex loss function of $\boldsymbol{p}_u$ and $\boldsymbol{q}_v$. In each SG iteration, we sample one entry to construct a subproblem problem combining a first-order approximation of the loss and the L2-regularization, the L1-regularization terms, and the two proximal terms $\|\boldsymbol{p}-\boldsymbol{p}_u\|_2^2$ and $\|\boldsymbol{q}-\boldsymbol{q}_v\|_2^2$. If an entry (u,v) is sampled, the corresponding subproblem is

$$\begin{aligned}&\min_{\boldsymbol{p}} \quad \boldsymbol{g}_u^T\boldsymbol{p} + \mu_p\|\boldsymbol{p}\|_1 + \frac{\sqrt{G_u}}{2\eta}\|\boldsymbol{p}-\boldsymbol{p}_u\|_2^2 \\ &\min_{\boldsymbol{q}} \quad \boldsymbol{h}_v^T\boldsymbol{q} + \mu_q\|\boldsymbol{q}\|_1 + \frac{\sqrt{H_v}}{2\eta}\|\boldsymbol{q}-\boldsymbol{q}_v\|_2^2 ,\end{aligned} \tag{2}$$

where

$$\begin{aligned}\boldsymbol{g}_u &= \partial_{\boldsymbol{p}_u}\phi_{u,v} = \partial_{\boldsymbol{p}_u} f + \lambda_p \boldsymbol{p}_u, \\ \boldsymbol{h}_v &= \partial_{\boldsymbol{q}_v}\phi_{u,v} = \partial_{\boldsymbol{q}_v} f + \lambda_q \boldsymbol{q}_v,\end{aligned}$$

and η is a pre-specified constant controlling the weight of the proximal terms. Obviously, the existence of the proximal terms encourages a new model close to the current model. See Table 1 for the supported loss functions and their subgradients $\partial_{\boldsymbol{p}} f$ and $\partial_{\boldsymbol{q}} f$. The two values G_u and H_v are respectively defined by

$$G_u = \frac{1}{k}\sum_{t=1}^{T_u-1}(\boldsymbol{g}_u^t)^T\boldsymbol{g}_u^t \text{ and } H_v = \frac{1}{k}\sum_{t=1}^{T_v-1}(\boldsymbol{h}_v^t)^T\boldsymbol{h}_v^t,$$

Table 1: Subgradients of supported loss functions in LIBMF. The first three are for RVMF, and the others are for BMF. Note we denote $\boldsymbol{p}^T\boldsymbol{q}$ by $\hat{r}$.

Loss	$f(\boldsymbol{p},\boldsymbol{q};r)$	$\partial_{\boldsymbol{p}} f$	$\partial_{\boldsymbol{q}} f$	κ
L2-norm	$(r-\hat{r})^2$	$\kappa\boldsymbol{q}$	$\kappa\boldsymbol{p}$	$\hat{r}-r$
L1-norm	$\lvert r-\hat{r}\rvert$	$\kappa\boldsymbol{q}$	$\kappa\boldsymbol{p}$	$\begin{cases}-1 & \text{if } r-\hat{r}>0\\ 1 & \text{if } r-\hat{r}<0\\ 0 & \text{otherwise}\end{cases}$
KL-divergence	$r\log(\frac{r}{\hat{r}})-r+\hat{r}$	$\kappa\boldsymbol{q}$	$\kappa\boldsymbol{p}$	$1-\frac{r}{\hat{r}}$
Squared hinge	$\max(0,1-r\hat{r})^2$	$\kappa\boldsymbol{q}$	$\kappa\boldsymbol{p}$	$-2r\max(0,1-r\hat{r})$
Hinge	$\max(0,1-\hat{r})$	$\kappa\boldsymbol{q}$	$\kappa\boldsymbol{p}$	$\begin{cases}-r & \text{if } 1-r\hat{r}>0\\ 0 & \text{otherwise}\end{cases}$
Logistic	$\log(1+\exp(-r\hat{r}))$	$\kappa\boldsymbol{q}$	$\kappa\boldsymbol{p}$	$\frac{-r\exp(-r\hat{r})}{1+\exp(-r\hat{r})}$

where T_u and T_v are the numbers of updates of $\boldsymbol{p}_u$ and $\boldsymbol{q}_v$, respectively. Notice that G_u and H_v are scalars rather than matrices in Duchi et al. (2011). Moreover, Duchi et al. (2011) use L2-norm instead of squared L2-norm in (2), and they do not provide the update rules considering both L1-regularization and L2-regularization. According to the optimality condition of (2), we can drive the SG update rules

$$\begin{aligned}(\boldsymbol{p}_u)_i &\leftarrow \operatorname{sign}\left((\boldsymbol{p}_u)_i - \frac{\eta}{\sqrt{G_u}}(\boldsymbol{g}_u)_i\right)\max\left(0, \lvert(\boldsymbol{p}_u)_i - \frac{\eta}{\sqrt{G_u}}(\boldsymbol{g}_u)_i\rvert - \frac{\eta}{\sqrt{G_u}}\mu_p\right)\\ (\boldsymbol{q}_v)_i &\leftarrow \operatorname{sign}\left((\boldsymbol{q}_v)_i - \frac{\eta}{\sqrt{H_v}}(\boldsymbol{h}_v)_i\right)\max\left(0, \lvert(\boldsymbol{q}_v)_i - \frac{\eta}{\sqrt{H_v}}(\boldsymbol{h}_v)_i\rvert - \frac{\eta}{\sqrt{H_v}}\mu_q\right),\end{aligned} \tag{3}$$

which can be reduced to

$$\begin{aligned}(\boldsymbol{p}_u)_i &\leftarrow (\boldsymbol{p}_u)_i - \frac{\eta}{\sqrt{G_u}}(\boldsymbol{g}_u)_i\\ (\boldsymbol{q}_v)_i &\leftarrow (\boldsymbol{q}_v)_i - \frac{\eta}{\sqrt{H_v}}(\boldsymbol{h}_v)_i\end{aligned}$$

if L1-regularization is dropped (i.e., $\mu_{\boldsymbol{p}}=\mu_{\boldsymbol{q}}=0$). The complete procedure of one SG iteration is shown in Algorithm 1. First, some variables are initialized and the value κ in Table 1 is computed and cached according to the selected loss function. We then apply (3) on all coordinates of $\boldsymbol{p}_u$ and $\boldsymbol{q}_v$, while G_u and H_v are updated in the end of the iteration. Note that for NMF we add two projections at line 16-19 by following Dror et al. (2012).

Algorithm 1 One SG update for solving RVMF and BMF in LIBMF.

$\bar{G} \leftarrow 0, \quad \bar{H} \leftarrow 0$
$\eta_u \leftarrow \eta_0/\sqrt{G_u}, \quad \eta_v \leftarrow \eta_0/\sqrt{H_v}$
$z \leftarrow \kappa(r_{u,v}, \boldsymbol{p}_u, \boldsymbol{q}_v)$
for $i \leftarrow 1$ to k **do**
  $(\boldsymbol{g}_u)_i \leftarrow \lambda_p(\boldsymbol{p}_u)_i + z(\boldsymbol{q}_v)_i$
  $(\boldsymbol{h}_v)_i \leftarrow \lambda_q(\boldsymbol{q}_v)_i + z(\boldsymbol{p}_u)_i$
  $\bar{G} \leftarrow \bar{G} + (\boldsymbol{g}_u)_i^2, \quad \bar{H} \leftarrow \bar{H} + (\boldsymbol{h}_v)_i^2$
  $(\boldsymbol{p}_u)_i \leftarrow (\boldsymbol{p}_u)_i - \eta_u(\boldsymbol{g}_u)_i$
  $(\boldsymbol{q}_v)_i \leftarrow (\boldsymbol{q}_v)_i - \eta_v(\boldsymbol{h}_v)_i$
  if $\mu_p > 0$ **then**
    $(\boldsymbol{p}_u)_i \leftarrow \operatorname{sign}((\boldsymbol{p}_u)_i) \max(0, |(\boldsymbol{p}_u)_i| - \eta_u\mu_p)$
  end if
  if $\mu_q > 0$ **then**
    $(\boldsymbol{q}_v)_i \leftarrow \operatorname{sign}((\boldsymbol{q}_v)_i) \max(0, |(\boldsymbol{q}_v)_i| - \eta_v\mu_q)$
  end if
  if NMF **then**
    $(\boldsymbol{p}_u)_i \leftarrow \max(0, (\boldsymbol{p}_u)_i)$
    $(\boldsymbol{q}_v)_i \leftarrow \max(0, (\boldsymbol{q}_v)_i)$
  end if
end for
$G_u \leftarrow G_u + \bar{G}/k, \quad H_v \leftarrow H_v + \bar{H}/k$

2. One-class Matrix Factorization in LIBMF

Recall the OCMF model (i.e., BPR) in LIBMF. BPR assumes that all positive entries should be ranked above all missing entries (i.e., dummy negative signals), so it is a kind of pairwise ranking problem. Naturally, the pairwise logistic loss can be used to compare the prediction values of a pair of positive and missing entries in the same row of R. We call the corresponding optimization problem (4) row-oriented BPR.

$$\begin{aligned}\min_{P,Q} \quad \sum_{(u,v)\in R} \sum_{(u,w)\notin R_u} \Big[&\log(1 + e^{\boldsymbol{p}_u^T(\boldsymbol{q}_w - \boldsymbol{q}_v)}) + \mu_p \|\boldsymbol{p}_u\|_1 + \\ &\mu_q(\|\boldsymbol{q}_v\|_1 + \|\boldsymbol{q}_w\|_1) + \frac{\lambda_p}{2}\|\boldsymbol{p}_u\|_2^2 + \frac{\lambda_q}{2}(\|\boldsymbol{q}_v\|_2^2 + \|\boldsymbol{q}_t\|_2^2)\Big],\end{aligned} \tag{4}$$

where R_u is the set of positive entries in the uth row of R. If u is column index and v and w are row indexes, then (4) becomes column-oriented BPR. If a row contains the ratings of a user, row-oriented BPR is suggested to build entry pairs user-wisely. Otherwise, column-oriented BPR should be used. Because we need a pair of entries to construct a subproblem for OCMF, the block scheduler in Chin et al. (2015a) is modified so that it returns two blocks in which we can select pairs of positive and a dummy negative entries at random. If a pair of a positive entry $(u, v) \in R$ and a negative entry $(u, w) \notin R$ is sampled, the

Algorithm 2 One SG update for solving OCMF in LIBMF.

$\bar{G}_u \leftarrow 0, \quad \bar{H}_v \leftarrow 0, \quad \bar{H}_w \leftarrow 0$
$\eta_u \leftarrow \eta_0/\sqrt{G_u}, \quad \eta_v \leftarrow \eta_0/\sqrt{H_w}, \quad \eta_w \leftarrow \eta_0/\sqrt{H_w}$
$z \leftarrow e^{\boldsymbol{p}_u^T(\boldsymbol{q}_w-\boldsymbol{q}_v)}/(1+e^{\boldsymbol{p}_u^T(\boldsymbol{q}_w-\boldsymbol{q}_v)})$
for $i \leftarrow 1$ to k **do**
  $(\boldsymbol{g}_u)_i \leftarrow \lambda_p(\boldsymbol{p}_u)_i - z((\boldsymbol{q}_w)_i - (\boldsymbol{q}_v)_i)$
  $(\boldsymbol{h}_v)_i \leftarrow \lambda_q(\boldsymbol{q}_v)_i - z(\boldsymbol{p}_u)_i$
  $(\boldsymbol{h}_w)_i \leftarrow \lambda_q(\boldsymbol{q}_w)_i + z(\boldsymbol{q}_u)_i$
  $\bar{G}_u \leftarrow \bar{G}_u + (\boldsymbol{g}_u)_i^2, \quad \bar{H}_v \leftarrow \bar{H}_v + (\boldsymbol{h}_v)_i^2, \quad \bar{H}_w \leftarrow \bar{H}_w + (\boldsymbol{h}_w)_i^2$
  $(\boldsymbol{p}_u)_i \leftarrow (\boldsymbol{p}_u)_i - \eta_u(\boldsymbol{g}_u)_i$
  $(\boldsymbol{q}_v)_i \leftarrow (\boldsymbol{q}_v)_i - \eta_v(\boldsymbol{h}_v)_i$
  $(\boldsymbol{q}_w)_i \leftarrow (\boldsymbol{q}_w)_i - \eta_w(\boldsymbol{h}_w)_i$
  if $\mu_p > 0$ **then**
    $(\boldsymbol{p}_u)_i \leftarrow \operatorname{sign}((\boldsymbol{p}_u)_i)\max(0, |(\boldsymbol{p}_u)_i| - \eta_u\mu_p)$
  end if
  if $\mu_q > 0$ **then**
    $(\boldsymbol{q}_v)_i \leftarrow \operatorname{sign}((\boldsymbol{q}_v)_i)\max(0, |(\boldsymbol{q}_v)_i| - \eta_v\mu_q)$
    $(\boldsymbol{q}_w)_i \leftarrow \operatorname{sign}((\boldsymbol{q}_w)_i)\max(0, |(\boldsymbol{q}_w)_i| - \eta_w\mu_q)$
  end if
  if NMF **then**
    $(\boldsymbol{p}_u)_i \leftarrow \max(0, (\boldsymbol{p}_u)_i)$
    $(\boldsymbol{q}_v)_i \leftarrow \max(0, (\boldsymbol{q}_v)_i)$
    $(\boldsymbol{q}_w)_i \leftarrow \max(0, (\boldsymbol{q}_w)_i)$
  end if
end for
$G_u \leftarrow G_u + \bar{G}_u/k, \quad H_v \leftarrow H_v + \bar{H}_v/k, \quad H_w \leftarrow H_w + \bar{H}_w/k$

subproblem of row-oriented BPR is

$$\begin{aligned}
&\min_{\boldsymbol{p}} \quad \boldsymbol{g}_u^T\boldsymbol{p} + \mu_p\|\boldsymbol{p}\|_1 + \frac{\sqrt{G_u}}{2\eta}\|\boldsymbol{p}-\boldsymbol{p}_u\|_2^2 \\
&\min_{\boldsymbol{q}} \quad \boldsymbol{h}_v^T\boldsymbol{q} + \mu_q\|\boldsymbol{q}\|_1 + \frac{\sqrt{H_v}}{2\eta}\|\boldsymbol{q}-\boldsymbol{q}_v\|_2^2 \,, \\
&\min_{\boldsymbol{q}} \quad \boldsymbol{h}_w^T\boldsymbol{q} + \mu_q\|\boldsymbol{q}\|_1 + \frac{\sqrt{H_w}}{2\eta}\|\boldsymbol{q}-\boldsymbol{q}_w\|_2^2
\end{aligned}$$

where

$$\begin{aligned}
\boldsymbol{g}_u &= z(\boldsymbol{q}_w - \boldsymbol{q}_v) + \lambda_p\boldsymbol{p}_u \\
\boldsymbol{h}_v &= -z\boldsymbol{p}_u + \lambda_q\boldsymbol{q}_v \\
\boldsymbol{h}_w &= z\boldsymbol{p}_u + \lambda_q\boldsymbol{q}_w
\end{aligned}$$

and

$$z = \frac{e^{\boldsymbol{p}_u^T(\boldsymbol{q}_w-\boldsymbol{q}_v)}}{1+e^{\boldsymbol{p}_u^T(\boldsymbol{q}_w-\boldsymbol{q}_v)}}.$$

We omit the update rules because they are the same as (3). Algorithm 2 shows the SG procedure for solving BPR.

References

Wei-Sheng Chin, Yong Zhuang, Yu-Chin Juan, and Chih-Jen Lin. A fast parallel stochastic gradient method for matrix factorization in shared memory systems. *ACM Transactions on Intelligent Systems and Technology*, 6:2:1–2:24, 2015a. URL `http://www.csie.ntu.edu.tw/~cjlin/papers/libmf/libmf_journal.pdf`.

Wei-Sheng Chin, Yong Zhuang, Yu-Chin Juan, and Chih-Jen Lin. A learning-rate schedule for stochastic gradient methods to matrix factorization. In *Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*, 2015b. URL `http://www.csie.ntu.edu.tw/~cjlin/papers/libmf/mf_adaptive_pakdd.pdf`.

Gideon Dror, Noam Koenigstein, Yehuda Koren, and Markus Weimer. The Yahoo! music dataset and KDD-Cup 11. In *JMLR Workshop and Conference Proceedings: Proceedings of KDD Cup 2011*, volume 18, pages 3–18, 2012.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12:2121–2159, 2011.